



**ISEL**

**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**  
**Área Departamental de Engenharia Mecânica**



## **Desenvolvimento de Aplicação para o Cálculo da Condução Térmica em Edifícios**

**JOÃO PEDRO FONSECA GONÇALVES**  
(Licenciado em Engenharia Mecânica)

Trabalho Final de Mestrado para obtenção do grau de Mestre  
em Engenharia Mecânica

**Orientadores:**

Doutora Cláudia Sofia Séneca da Luz Casaca  
Mestre Fernando Paulo Neves da Fonseca Cardoso Carreira

**Júri:**

Presidente: Doutor João Manuel Ferreira Calado

Vogais:

Doutor Pedro Miguel Abreu Silva  
Doutora Cláudia Sofia Séneca da Luz Casaca

**Dezembro de 2016**





**ISEL**

**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**  
**Área Departamental de Engenharia Mecânica**



## **Desenvolvimento de Aplicação para o Cálculo da Condução Térmica em Edifícios**

**JOÃO PEDRO FONSECA GONÇALVES**  
(Licenciado em Engenharia Mecânica)

Trabalho Final de Mestrado para obtenção do grau de Mestre  
em Engenharia Mecânica

**Orientadores:**

Doutora Cláudia Sofia Séneca da Luz Casaca  
Mestre Fernando Paulo Neves da Fonseca Cardoso Carreira

**Júri:**

Presidente: Doutor João Manuel Ferreira Calado

Vogais:

Doutor Pedro Miguel Abreu Silva  
Doutora Cláudia Sofia Séneca da Luz Casaca

**Dezembro de 2016**



## **Agradecimentos**

Gostaria de agradecer à Professora Cláudia Casaca e ao Professor Fernando Carreira pela excelência na sua orientação e pelos conhecimentos transmitidos.

Agradeço aos meus pais e à minha irmã Inês Gonçalves pelo apoio incondicional, confiança depositada e pela paciência demonstrada na realização desta dissertação.



## Resumo

A presente dissertação consistiu no desenvolvimento de uma aplicação para o cálculo da condução térmica em edifícios. Esta aplicação foi desenvolvida através do programa de cálculo computacional MATrix LABoratory (MATLAB). Também é permitida a introdução e visualização dos dados relativos aos elementos estruturais, sendo feita de forma intuitiva para o utilizador. As estruturas de dados permitem manipular a informação relativa a materiais de construção de edifícios, locais geográficos, vidros simples, duplos e triplos, caixilhos, janelas simples e duplas, portas, elementos construtivos, horários de ocupação, iluminação e de equipamentos eléctricos. Para além da estrutura de dados a aplicação é composta por 18 ambientes gráficos e 19 funções que permitem: i) criar projectos, ii) introduzir e visualizar informação complementar à estrutura de dados original da aplicação, iii) definir e visualizar espaços a serem analisados termicamente, iv) introduzir, visualizar e definir os diversos tipos de horários, v) exportar os dados relativos à estrutura de dados e os relatórios da análise térmica do espaço.

Os cálculos que a aplicação desenvolvida efectua são os das cargas térmicas de aquecimento e arrefecimento de acordo com as condições de projecto. A aplicação desenvolvida efectua também o cálculo das mesmas cargas térmicas em cada hora do ano, através da importação de um ficheiro climático. Desse ficheiro são retiradas as informações necessárias do clima do local geográfico em análise. Na formulação matemática é apresentada a metodologia de cálculo dos coeficientes globais de transmissão de calor essenciais para o cálculo das cargas térmicas, das próprias cargas térmicas e de conceitos psicométricos.

A comparação dos resultados obtidos pela aplicação desenvolvida nesta dissertação foi realizada com uma outra aplicação de referência. Esta comparação permitiu verificar algumas discrepâncias no cálculo das cargas térmicas. Nomeadamente, no cálculo das cargas térmicas dos elementos da envolvente exterior. Uma vez que o programa utilizado para a comparação, como outras aplicações que calculam as cargas térmicas, consideram outros fenómenos físicos, não considerados neste estudo. No caso em que são considerados os mesmos fenómenos físicos a aplicação desenvolvida apresenta valores com um erro associado inferior a 1%.

**Palavras-chave:** Transferência de calor; Cargas térmicas; Aplicações de cálculo de cargas térmicas; MATLAB.





## Abstract

This master thesis consisted on the development of an application for the calculation of thermal conduction in buildings. This was developed by the computational calculation software MATrix LABoratory (MATLAB). It allows the introduction and display of data on structural elements in a user-friendly layout. Databases allow to manipulate the information on buildings' construction materials, geographic locations, single, double and triple glazing, frames, single and double windows, doors, construction elements, occupancy schedules, lighting and electrical equipment. Apart from the database this application consists of 18 graphical environments and 19 functions that allow: i) to create projects, ii) to introduce and visualize complementary information to the original data base, iii) to define and visualize areas to be analyzed thermally iv) introduce, visualize and define the several different schedules, v) to export the database data and the thermal analysis reports of the area in study.

The calculations that this application performs are the calculation of thermal loads of heating and cooling according to the designed conditions. Also, it is able to calculate the same thermal loads at every time of the year, by importing a climate file. From that file it is taken the necessary climate information of the geographical location in analysis. In the mathematical formulation it is presented the calculation methodology of the overall coefficient of heat transfer essential for the calculation of thermal loads, thermal loads itself and psychometric concepts.

The results obtained by this application developed in this project were compared to another reference application. The comparison of results enabled to evaluate some discrepancies in the calculation of thermal loads. In particular, the calculation of thermal loads of the outer surrounding elements. Since the program used for the comparison, as well as other applications that calculate thermal loads, another physical phenomena's are considered which were not in this study. When the same physical phenomena are taken into consideration, the associated error is inferior to 1% for the developed application.

**Keywords:** Heat transfer; Thermal loads; Software for thermal loads calculation; MATLAB.



## Lista de símbolos

$A$	Área	$[m^2]$
$A_f$	Área do perfil da caixilharia	$[m^2]$
$A_{f,di}$	Área de desenvolvimento interno do caixilho	$[m^2]$
$A_{f,de}$	Área de desenvolvimento externo do caixilho	$[m^2]$
$A_{f,e}$	Área projectada exterior do caixilho	$[m^2]$
$A_{f,i}$	Área projectada interior do caixilho	$[m^2]$
$A_g$	Área do vidro	$[m^2]$
$A_p$	Área do painel opaco	$[m^2]$
$A_w$	Área da Janela	$[m^2]$
$b_f$	Largura em projecção da caixilharia	$[m]$
$b_p$	Largura visível do painel	$[m]$
$d$	Espessura do material	$[m]$
$d_v$	Espessura do vidro	$[m]$
$d_n$	Densidade de ocupação	$[m^2/\text{pessoa}]$
$E_M$	Eficiência do motor	
$E_n$	Poder emissivo	$[W/m^2]$
$F_{ul}$	Factor de utilização	
$F_{sa}$	Factor de compensação	
$F_{UM}$	Factor de utilização do motor	
$F_{LM}$	Factor de carga do motor	
$F_L$	Factor de carga da aplicação	
$g_0$	Constante de aceleração gravitacional	$[m/s^2]$
$H$	Altura sobre o nível do mar	$[m]$
$h$	Entalpia do ar	$[kJ/kg]$
$h_a$	Entalpia do ar seco	$[kJ/kg]$
$H_b$	Altura na camada inferior da atmosfera	$[m]$
$h_{in}$	Entalpia do ar interior	$[kJ/kg]$
$h_{out}$	Entalpia do ar exterior	$[kJ/kg]$
$h_v$	Entalpia do vapor de água	$[kJ/kg]$
$h_\infty$	Coeficiente de convecção	$[W/(m^2.K)]$
$I_g$	Perímetro total do vidro	$[m]$

$I_p$	Perímetro do painel opaco	[m]
$k$	Condutibilidade térmica	[W/(m.K)]
$L$	Espessura	[m]
$L_b$	Taxa de lapso da temperatura	[K/m]
$L_f^{2D}$	Transmissão de calor bidimensional através da secção da caixilharia	[W/(m.K)]
$M$	Massa molar do ar	[kg/mol]
$N$	Número de ocupantes	
$P$	Potência do motor	[W]
$p$	Pressão atmosférica	[Pa]
$P_b$	Pressão estática (ao nível do mar)	[Pa]
$p_s$	Pressão de saturação	[Pa]
$p_v$	Pressão parcial do vapor	[Pa]
$q_{ar,l}$	Carga latente devido à ventilação e infiltração de ar	[W]
$q_{ar,s}$	Carga sensível devido à ventilação e infiltração de ar	[W]
$q_{ar,total}$	Carga total devido à ventilação e infiltração de ar	[W]
$q_{cpc}$	Carga térmica devido à envolvente exterior	[W]
$q_{div,l}$	Carga térmica latente devido a aplicações diversas	[W]
$q_{div,s}$	Carga sensível da aplicação	[W]
$q_{dtc}$	Carga térmica devido à envolvente interior	[W]
$q_{env,ext}$	Ganho de calor através da envolvente externa	[W]
$q_{env,int}$	Ganho de calor através da envolvente interna	[W]
$q_{eq,ele}$	Carga devida a equipamentos com motor eléctrico	[W]
$q_{ilu}$	Carga térmica devido à iluminação	[W]
$q_{input}$	Energia trocada pela aplicação	[W]
$q_{l,per}$	Carga latente por ocupante	[W/pessoa]
$q_{ocup,l}$	Carga latente devida aos ocupantes	[W]
$q_{ocup,s}$	Carga sensível devida aos ocupantes	[W]
$q_p$	Carga térmica devido a equipamentos eléctricos	[W]
$q_s''$	Fluxo de calor por convecção	[W/m <sup>2</sup> ]
$q_{s,per}$	Carga sensível por ocupante	[W/pessoa]
$q_{total}$	Carga térmica de arrefecimento / aquecimento	[W]
$q_{total,l}$	Carga térmica de arrefecimento / aquecimento latente	[W]

$q_{\text{total},s}$	Carga térmica de arrefecimento / aquecimento sensível	[W]
$q_x$	Fluxo de calor por condução	[W]
$R$	Constante universal de um gás	[(N.m)/(mol.K)]
$R_{se}$	Resistência térmica exterior	[(m <sup>2</sup> .K)/W]
$R_{si}$	Resistência térmica interior	[(m <sup>2</sup> .K)/W]
$R_s$	Resistência térmica do espaço entre o vidro interior e exterior	[(m <sup>2</sup> .K)/W]
$T$	Temperatura	[°C]
$T_b$	Temperatura média do ar do espaço adjacente	[°C]
$T_{bm}$	Temperatura normalizada (ao nível do mar)	[K]
$T_{in}$	Temperatura interior	[°C]
$T_{out}$	Temperatura exterior	[°C]
$T_{su}$	Temperatura da superfície	[K]
$T_s^4$	Temperatura absoluta	[K]
$T_1$	Temperatura da superfície 1	[K]
$T_2$	Temperatura da superfície 2	[K]
$T_\infty$	Temperatura do fluído	[K]
$U$	Coeficiente global de transmissão de calor	[W/(m <sup>2</sup> .K)]
$U_D$	Coeficiente global de transmissão de calor da porta	[W/(m <sup>2</sup> .K)]
$U_f$	Coeficiente global de transmissão de calor do caixilho	[W/(m <sup>2</sup> .K)]
$U_g$	Coeficiente global de transmissão de calor do vidro simples	[W/(m <sup>2</sup> .K)]
$U_p$	Coeficiente global de transmissão de calor do painel opaco	[W/(m <sup>2</sup> .K)]
$U_w$	Coeficiente global de transmissão de calor da janela	[W/(m <sup>2</sup> .K)]
$U_{ws}$	Coeficiente global de transmissão de calor da janela com dispositivo de oclusão fechado	[W/(m <sup>2</sup> .K)]
$\dot{V}$	Caudal volúmico de ar	[m <sup>3</sup> /s]
$W$	Potência total de iluminação	[W]
$W_{1A}$	Densidade de iluminação	[W/m <sup>2</sup> ]
$\Delta R$	Resistência térmica adicional devido à camada de ar entre o dispositivo de oclusão e a janela	[(m <sup>2</sup> .K)/W]
$\lambda$	Condutibilidade térmica do material	[W/(m.K)]
$\lambda_v$	Condutibilidade térmica do vidro	[W/(m.K)]
$\sigma$	Constante de Stefan-Boltzmann	[W/(m <sup>2</sup> .K <sup>4</sup> )]
$\Psi_g$	Coeficiente de transmissão térmica linear do vidro	[W/(m.K)]

$\Psi_p$	Coeficiente de transmissão térmica linear do painel opaco	[W/(m.K)]
$\omega$	Humidade específica	[kg <sub>v</sub> /kg <sub>a</sub> ]
$\omega_{in}$	Humidade específica do ar interior	[kg <sub>v</sub> /kg <sub>a</sub> ]
$\omega_{out}$	Humidade específica do ar exterior	[kg <sub>v</sub> /kg <sub>a</sub> ]
$\frac{\partial T}{\partial x}$	Gradiente Térmico no sentido do fluxo de calor	[K/m]

## Lista de abreviaturas e acrónimos

<b>ASHRAE</b>	Sociedade Americana de Engenheiros de Aquecimento, Refrigeração e Ar-condicionado ou <i>American Society of Heating, Refrigerating and Air-Conditioning Engineers</i>
<b>AVAC</b>	Aquecimento, Ventilação e Ar Condicionado
<b>BDL</b>	<i>Building Description Language</i>
<b>BLAST</b>	<i>Building Load Analysis and System Thermodynamics</i>
<b>CLTD/SCL/CLF</b>	Método da diferença de temperatura ou <i>Cooling Load Temperature Difference/ Solar Cooling Load Factor/ Cooling Load Factor</i>
<b>CTF</b>	Coeficientes condutivos da função de transferência ou <i>Conduction Transfer Function coefficients</i>
<b>DAC</b>	Desenho assistido por computador
<b>DOE</b>	Departamento de energia ou <i>Department of Energy</i>
<b>GUI</b>	Interface gráfica de utilizador ou <i>Graphical User Interface</i>
<b>HBM</b>	Método do balanço térmico ou <i>Heat Balance Method</i>
<b>HAP</b>	<i>Hourly Analysis Program</i>
<b>IES</b>	<i>Integrated Environmental Solutions</i>
<b>ITE</b>	Informação Técnica de Edifícios
<b>JJH</b>	<i>James J.Hirsch &amp; Associates</i>
<b>MATLAB</b>	<i>MATrix LABoratory</i>
<b>NBSLD</b>	<i>National Bureau of Standards Load Determination</i>
<b>LBNL</b>	Laboratório Nacional de Lawrence Berkeley ou <i>Lawrence Berkeley National Laboratory</i>
<b>LNEC</b>	Laboratório Nacional de Engenharia Civil
<b>RTF</b>	Coeficientes de zona de função de transferência ou <i>coefficients of Room Transfer Function</i>
<b>RTS</b>	Método das séries radiactivas ou <i>Radiant Time Series</i>
<b>TETD/TA</b>	Método da diferença de temperatura diferencial ou <i>Total Equivalent Temperature diferencial/Time Averaging</i>
<b>TFM</b>	Método das funções de transferência ou <i>Transfer Function Method</i>
<b>TRACE</b>	<i>Trane Air Conditioning Economics</i>
<b>TRNSYS</b>	<i>Transient System Simulation Program</i>





# Índice

Agradecimentos.....	i
Resumo.....	iii
Abstract .....	v
Lista de símbolos.....	vii
Lista de abreviaturas e acrónimos .....	xi
Índice de Tabelas.....	xvii
Índice de Figuras .....	xix
1. Introdução.....	1
1.1. Objectivos.....	2
1.2. Estrutura global da dissertação .....	2
2. Revisão Bibliográfica.....	5
2.1. Transferência de Calor.....	5
2.1.1. Condução.....	5
2.1.2. Convecção .....	7
2.1.3. Radiação.....	8
2.2. Metodologias de cálculo de cargas térmicas em programas informáticos .....	9
2.2.1. Método do balanço térmico.....	10
2.2.2. Método da diferença de temperatura diferencial.....	11
2.2.3. Método das funções de transferência .....	12
2.2.4. Método da diferença de temperatura.....	12
2.2.5. Método das séries radiactivas.....	13
2.3. Programas informáticos de simulação .....	14
2.3.1. Hourly Analysis Program.....	14
2.3.2. EnergyPlus .....	16
2.3.3. DOE-2 .....	17
2.3.4. Integrated Environmental Solutions.....	17
2.3.5. Transient System Simulation Program.....	18
2.3.6. Trane Air Conditioning Economics .....	19
2.4. Soluções construtivas .....	20
2.4.1. Paredes de fachada .....	20
2.4.2. Pavimentos .....	21
2.4.3. Coberturas .....	22
2.4.4. Vão envidraçado.....	24
2.4.5. Caixilharia .....	26

2.4.6. Vidro.....	26
3. Formulação Matemática .....	29
3.1. Coeficiente global de transmissão de calor da envolvente opaca.....	29
3.2. Vãos envidraçados .....	30
3.2.1. Definição Geométrica do envidraçado .....	30
3.2.2. Vidro simples e duplo .....	32
3.2.3. Perfis de caixilharia.....	33
3.2.4. Janela simples.....	34
3.2.5. Janela dupla .....	36
3.2.6. Janela acoplada.....	37
3.2.7. Janela com dispositivo de oclusão fechado.....	38
3.2.8. Portas .....	38
3.3. Cargas térmicas.....	39
3.3.1. Ganho de calor condutivo através da envolvente exterior .....	39
3.3.2. Ganho de calor através da envolvente interior .....	40
3.3.3. Ocupação .....	40
3.3.4. Iluminação .....	42
3.3.5. Carga devida a equipamentos com motor eléctrico .....	42
3.3.6. Carga devida a aplicações diversas .....	43
3.3.7. Carga devida à infiltração de ar.....	44
3.3.8. Carga térmica de arrefecimento e aquecimento .....	45
3.4. Psicrometria .....	46
3.4.1. Humidade relativa .....	46
3.4.2. Humidade específica .....	47
3.4.3. Entalpia.....	48
4. Programação desenvolvida.....	49
4.1. Estruturas de dados .....	49
4.1.1. Materiais de construção de edifícios .....	50
4.1.2. Locais .....	51
4.1.3. Vidros simples, duplos e triplos .....	52
4.1.4. Caixilhos.....	52
4.1.5. Janelas .....	53
4.1.6. Portas.....	53
4.1.7. Elementos construtivos: paredes de fachada, pavimentos e coberturas .....	53
4.1.8. Horários de ocupação, iluminação e de equipamentos eléctricos .....	54

4.2.	Ambiente gráfico .....	55
4.2.1.	Gui principal.....	56
4.2.2.	GUI de definição do local .....	58
4.2.3.	GUI de Materiais .....	60
4.2.4.	GUI de vidros .....	61
4.2.5.	GUI de caixilho .....	64
4.2.6.	GUI de adicionar janela simples .....	66
4.2.7.	GUI de adicionar janela dupla.....	68
4.2.8.	GUI de adicionar janela de dupla folha no mesmo aro .....	69
4.2.9.	GUI de Ver/Eliminar Janela .....	70
4.2.10.	GUI de adicionar porta .....	71
4.2.11.	GUI de Ver/Eliminar Porta.....	72
4.2.12.	GUI de adicionar parede, tecto, cobertura e pavimento .....	73
4.2.13.	GUI de Ver/Eliminar Parede, Tecto, Cobertura e Pavimento .....	75
4.2.14.	GUI de Adicionar Espaço.....	76
4.2.15.	GUI de Ver/Eliminar Espaço.....	78
4.2.16.	GUI de Adicionar Horário .....	80
4.2.17.	GUI de Definir Horário .....	81
4.2.18.	GUI de Relatórios .....	82
5.	Validação da aplicação.....	87
5.1.	Caso de estudo .....	87
5.2.	Comparação de resultados .....	91
6.	Conclusões e desenvolvimentos futuros .....	99
	Referências Bibliográficas .....	103
	Anexos.....	105



## Índice de Tabelas

Tabela 3.1. Resistências Térmicas Superficiais [3].	30
Tabela 3.2. Resistências térmicas de espaços de ar não ventilados para janelas duplas ou acopladas [21].	33
Tabela 3.3. Calor sensível e latente de ocupantes	41
Tabela 3.4. Factores de carga de algumas cargas diversas [7].	44
Tabela 4.1. Descrição da estrutura de dados	50
Tabela 4.2. Níveis de menus da aplicação.	56
Tabela 5.1. Parede exterior 1.	88
Tabela 5.2. Parede exterior 2.	89
Tabela 5.3. Cobertura do espaço.	90
Tabela 5.4. Parede interior 1.	90
Tabela 5.5. Parede interior 2.	90
Tabela 5.6. Cargas térmicas de arrefecimento.	92
Tabela 5.7. Cargas térmicas de aquecimento.	95



# Índice de Figuras

Figura 2.1. Transferência de calor unidimensional por condução [4].	7
Figura 2.2. Processos de transferência de calor por convecção: (a) Convecção forçada; (b) Convecção natural [4].	8
Figura 2.3. Janela inicial do HAP.	15
Figura 2.4. Energy Plus – Janela “Início” [11].	16
Figura 2.5. Simulation Studio TRNSYS [13].	18
Figura 2.6. Project Navigator, TRACE 700.	19
Figura 2.7. Camadas genéricas constituintes de um pavimento [17].	21
Figura 2.8. Impermeabilização de pavimento térreo [16].	21
Figura 2.9. Cobertura inclinada com desvão não habitável – Isolamento térmico na esteira horizontal [19].	23
Figura 2.10. Cobertura inclinada com desvão habitável – Isolamento térmico nas vertentes [19].	24
Figura 2.11. Definição dos elementos da janela [20].	25
Figura 2.12. Definição dos elementos da porta [20].	25
Figura 2.13. Representação de um vidro de isolamento térmico (vidro duplo) [20].	27
Figura 3.1. Perímetro e área do vidro [21].	31
Figura 3.2. Representação das diversas áreas de um vão envidraçado [21].	31
Figura 3.3. Área de desenvolvimento do caixilho [21].	32
Figura 3.4. Caixilharia com painel opaco [22].	33
Figura 3.5. Representação esquemática de uma janela simples [21].	34
Figura 3.6. Representação esquemática de uma janela dupla [21].	36
Figura 3.7. Representação esquemática de uma janela acoplada [21].	37
Figura 3.8. Janela com dispositivo de oclusão fechado [21].	38
Figura 4.1. Primeiro elemento da classe de betão na estrutura de dados de materiais de construção.	51
Figura 4.2. Primeiro elemento na estrutura de dados de locais.	51
Figura 4.3. Primeiro elemento da estrutura de dados de vidros.	52
Figura 4.4. Primeiro elemento da estrutura de dados de caixilhos.	52
Figura 4.5. Primeiro elemento da estrutura de dados de janelas.	53
Figura 4.6. Primeiro elemento da estrutura de dados de portas.	53
Figura 4.7. Primeiro elemento da estrutura de dados de elementos construtivos.	54
Figura 4.8. Elementos na estrutura de dados de ocupação, iluminação e equipamentos eléctricos.	55
Figura 4.9. GUI principal.	57
Figura 4.10. GUI de definição do local	59
Figura 4.11. GUI de adicionar, ver e eliminar materiais.	60
Figura 4.12. GUI de adicionar vidros.	62
Figura 4.13. GUI de ver / eliminar vidros.	63
Figura 4.14. GUI de adicionar caixilho.	65
Figura 4.15. GUI de ver / eliminar caixilho.	66
Figura 4.16. GUI de adicionar janela simples.	67
Figura 4.17. GUI de adicionar janela dupla.	68

Figura 4.18. GUI de adicionar janela de dupla folha no mesmo aro.....	69
Figura 4.19. GUI de Ver / Eliminar Janela. ....	71
Figura 4.20. GUI de adicionar porta. ....	72
Figura 4.21. GUI de ver / eliminar porta. ....	73
Figura 4.22. GUI de adicionar parede, tecto, cobertura e pavimento. ....	74
Figura 4.23. GUI de Ver / Eliminar Parede, Tecto, Cobertura e Pavimento.....	76
Figura 4.24. GUI de adicionar espaço.....	77
Figura 4.25. GUI de ver / eliminar espaço. ....	79
Figura 4.26. GUI de adicionar horário. ....	80
Figura 4.27. GUI de definir horário .....	82
Figura 4.28. GUI de relatórios. ....	83
Figura 5.1. Representação em planta do espaço definido como caso de estudo. ....	88
Figura 5.2. Cargas de arrefecimento – Envolvente exterior.....	93
Figura 5.3. Cargas de arrefecimento dos diferentes tipos de cargas térmicas.....	94
Figura 5.4. Cargas de aquecimento – Envolvente exterior. ....	96
Figura 5.5. Carga Térmica total – Análise horária. ....	96



# 1. Introdução

Actualmente, a maioria dos cálculos das cargas térmicas são efectuados por computadores, com recurso a programas informáticos próprios para o efeito. Visto a existência de uma necessidade acrescida da utilização de ferramentas que simplifiquem ou reduzam o tempo de cálculo de projectos. A temática abordada nesta dissertação é bastante relevante, visto que existe uma maior preocupação com as questões ambientais, conforto térmico, eficiência energética, e aspectos económicos.

Os programas de cálculo de cargas térmicas podem ser divididos em duas categorias. A primeira categoria inclui os programas desenvolvidos por Instituições Públicas ou Universidades. Como segunda categoria temos os programas que são desenvolvidos por empresas do sector privado. As aplicações comerciais não são gratuitas e são disponibilizadas com o código compilado. Isto leva que, usualmente não se conhece o código utilizado nem os modelos matemáticos utilizados para a realização dos cálculos. Como desvantagens as aplicações comerciais têm:

- Modelo matemático não pode ser alterado;
- Modelo matemático normalmente não é conhecido;
- Têm estruturas de dados sem informação de materiais e tipologias de construção nacionais.

Desenvolver uma aplicação para o cálculo da condução térmica em edifícios em código aberto é relevante porque permite a qualquer utilizador que saiba programar personalizar a sua aplicação ou até mesmo modificar ou introduzir novos cálculos ou funcionalidades ao programa. A aplicação desenvolvida é de código aberto e de livre acesso, tendo uma vasta estrutura de dados: materiais de construção de edifícios, locais, vidros simples, duplos e triplos, caixilhos, janelas simples e duplas, portas, elementos construtivos e horários de ocupação, iluminação e de equipamentos eléctricos. Podendo potencializar e adequar estas estruturas de dados na sua maioria com informação de materiais e tipologias de construção nacionais.

## **1.1. Objectivos**

Como objectivos deste trabalho final de mestrado foram definidos que deveria ser realizado o estudo e construção de um modelo matemático para o cálculo da condução térmica em edifícios, ao longo do ano. O modelo deverá ter em conta uma vasta estrutura de dados no âmbito dos vários tipos de construção de edifícios, constituição de paredes, pavimentos, coberturas, portas e janelas. Adicionalmente, deverá ser possível calcular o calor transmitido por condução em todas as divisões do edifício.

Complementarmente foi pretendido desenvolver uma aplicação de uso fácil e intuitivo na introdução de dados e recepção de informação sobre o espaço ou espaços em estudo. Para tal é utilizado o ambiente gráfico do MATLAB, e são criados vários menus e submenus para uma melhor navegação na aplicação. Ou seja, são criadas várias janelas em ambiente gráfico. O desenvolvimento da aplicação permite que a disposição das janelas não se encontre sobreposta, logo visualmente toda a aplicação funciona numa única janela.

Numa última fase pretende-se elaborar uma comparação entre os valores obtidos: da aplicação desenvolvida em MATLAB, de uma aplicação de referência e do método analítico (utilizando a mesma metodologia de cálculo da aplicação desenvolvida). A comparação entre os valores obtidos da aplicação desenvolvida e do método analítico servirá como verificação da existência de algum erro associado à programação desenvolvida.

## **1.2. Estrutura global da dissertação**

No primeiro capítulo é descrita a estrutura global da dissertação, são descritos os objectivos definidos para este trabalho final de mestrado e referida a relevância do tema.

O segundo capítulo é denominado de Revisão Bibliográfica, e são abordadas as temáticas: da transferência de calor (que é a base do cálculo das cargas térmicas), das metodologias de cálculo de cargas térmicas em programas informáticos, programas informáticos de simulação e de algumas soluções construtivas de paredes de fachada, pavimentos, coberturas, vãos envidraçados, caixilharias e vidros. A formulação matemática apresenta a metodologia de cálculo dos coeficientes globais de transmissão de calor essenciais para o cálculo das cargas térmicas, das próprias cargas térmicas e de conceitos psicométricos. Sendo a formulação matemática desenvolvida no terceiro capítulo desta dissertação.

O quarto capítulo é designado de Programação desenvolvida. Neste capítulo é abordada em detalhe o desenvolvimento da aplicação informática para o cálculo da condução térmica em edifícios. A aplicação é constituída por três etapas: i) construção das estruturas de dados, ii) elaboração de funções, iii) construção da interface gráfica. Foram abordados os vários tipos de estruturas de dados constituintes da aplicação como também foram abordadas as várias interfaces gráficas criadas. Durante a apresentação do desenvolvimento das interfaces gráficas são introduzidas as funções associadas aos objectos existentes em cada uma. No quinto capítulo foi realizada uma comparação dos valores obtidos através do cálculo analítico, dos valores da aplicação informática desenvolvida e do programa *Hourly Analysis Program* (HAP). Finalmente, no Capítulo 6, as principais conclusões do trabalho são sintetizadas e algumas recomendações para trabalho futuro são enumeradas.



## 2. Revisão Bibliográfica

### 2.1. Transferência de Calor

A transferência de calor é a ciência que pretende prever o fluxo de energia resultante de uma diferença de temperatura. Esta ciência não procura apenas explicar como a energia calorífica é transferida, mas também prever a taxa de permuta [1]. Na permuta de calor existem três formas distintas, sendo elas a condução, a convecção e a radiação. Na prática, as três formas de transmissão ocorrem, quase sempre, simultaneamente. Porém, o estudo faz-se sobre as formas predominantes, desprezando-se uma ou duas delas [2].

#### 2.1.1. Condução

Segundo Holman (2010), quando existe um gradiente de temperatura num corpo, a energia é transferida da região com a temperatura mais elevada para a região com temperatura mais baixa. Assim sendo, a energia é transferida por condução quando o fluxo de calor transferido por unidade de área é proporcional ao gradiente de temperatura normal [1]:

$$\frac{q_x}{A} \sim \frac{\partial T}{\partial x} \quad (2.1)$$

Em que:

$q_x$  – Fluxo de calor por condução [W];

$A$  – Área [m<sup>2</sup>];

$\frac{\partial T}{\partial x}$  – Gradiente térmico no sentido do fluxo de calor [K/m].

Para quantificar o processo de condução em termos de equações de fluxo, existe a Lei de Fourier. Esta Lei é uma relação vectorial uma vez que tem sentido, orientação e módulo. Introduzindo a constante de proporcionalidade  $k$  na equação (2.1), a lei de Fourier é definida como:

$$q_x = -kA \frac{\partial T}{\partial x} \quad (2.2)$$

Em que:

$q_x$  – Fluxo de calor por condução [W];

$k$  – Condutibilidade térmica [W/(m.K)];

$A$  – Área [m<sup>2</sup>];

$\frac{\partial T}{\partial x}$  – Gradiente térmico no sentido do fluxo de calor [K/m].

A constante de proporcionalidade,  $k$ , é designada por condutibilidade térmica e é uma característica do material. A condutibilidade térmica é uma propriedade que caracteriza os materiais ou produtos termicamente homogêneos, representando a quantidade de calor que atravessa uma espessura unitária de um material, quando entre duas faces planas e paralelas se estabelece uma diferença unitária de temperatura [3]. O sinal negativo da equação (2.2) é a consequência de o calor ser transferido na direcção da menor temperatura.

Em condições estacionárias, como demonstrada na Figura 2.1, com a distribuição de temperaturas linear, o gradiente de temperatura pode ser definido como:

$$\frac{\partial T}{\partial x} = \frac{T_2 - T_1}{L} \quad (2.3)$$

Em que:

$\frac{\partial T}{\partial x}$  – Gradiente térmico no sentido do fluxo de calor [K/m];

$T_1$  – Temperatura da superfície 1 [K];

$T_2$  – Temperatura da superfície 2 [K];

$L$  – Espessura [m].

Substituindo na equação (2.2) a expressão simplificada do gradiente de temperatura, equação (2.3), o fluxo de calor em condições estacionárias com a distribuição de temperaturas linear, é definido da seguinte forma:

$$q_x = -kA \frac{T_1 - T_2}{L} \quad (2.4)$$

Em que:

$q_x$  – Fluxo de calor por condução [W];

$k$  – Condutibilidade térmica [W/(m.K)];

$A$  – Área [m<sup>2</sup>];

$T_1$  – Temperatura da superfície 1 [K];

$T_2$  – Temperatura da superfície 2 [K];

$L$  – Espessura [m].

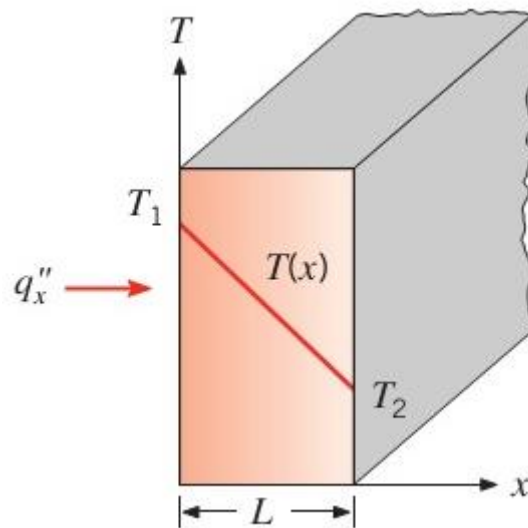


Figura 2.1. Transferência de calor unidimensional por condução [4].

### 2.1.2. Convecção

A convecção ocorre quando existe contacto entre um fluido em movimento e uma superfície a diferentes temperaturas. A transferência de calor por convecção pode ser classificada de acordo com a natureza do escoamento do fluido. Referimo-nos a convecção forçada quando o escoamento é causado por meios externos, tais como um ventilador, uma bomba, ou ventos atmosféricos. Em contraste, no caso da convecção livre (ou natural) o escoamento do fluido é induzido por forças que são originadas a partir de diferenças de massa específica causadas por variações de temperatura no fluido [4]. A Figura 2.2 representa os processos de convecção forçada e natural.

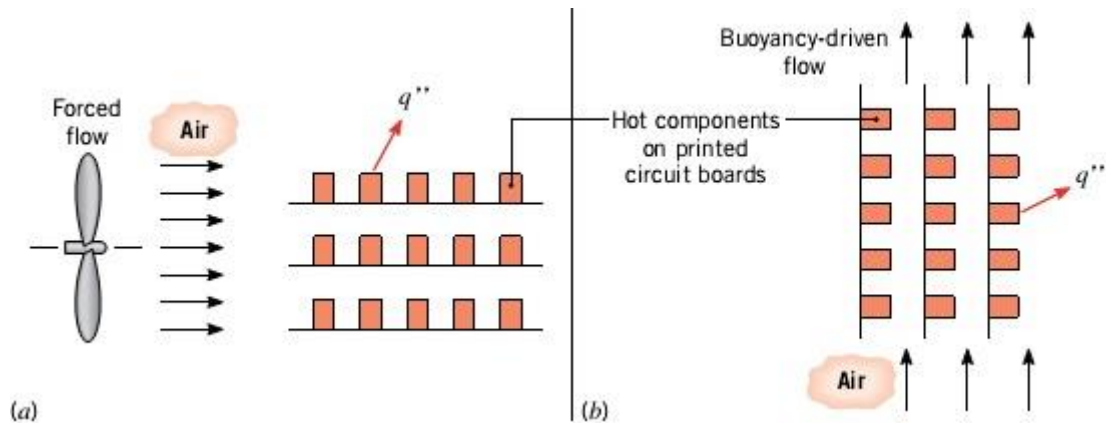


Figura 2.2. Processos de transferência de calor por convecção: (a) Convecção forçada; (b) Convecção natural [4].

Independentemente da natureza específica do processo de transferência de calor por convecção, a equação apropriada para a taxa de transferência possui a seguinte forma:

$$q_s'' = h_\infty (T_{su} - T_\infty) \quad (2.5)$$

Em que:

$q_s''$  – Fluxo de calor por convecção [ $\text{W}/\text{m}^2$ ];

$h_\infty$  – Coeficiente de convecção [ $\text{W}/(\text{m}^2\text{K})$ ];

$T_{su}$  – Temperatura da superfície [K];

$T_\infty$  – Temperatura do fluído [K].

### 2.1.3. Radiação

A radiação é a energia emitida pela matéria que se encontra a uma temperatura diferente de zero. Este fenómeno de transferência de energia pode ocorrer a partir de superfícies sólidas, mas também a partir de gases e líquidos. Enquanto a transferência de energia através da condução ou convecção necessitam de um meio, a radiação não requer de nenhum meio, uma vez que a radiação é transferida por ondas electromagnéticas emitidas pela fonte de calor [4].



Para quantificar a taxa de energia libertada devido à radiação por unidade de área existe uma propriedade denominada por poder emissivo,  $E_n$ , de uma superfície. O poder emissivo tem um limite superior, sendo ele determinado pela lei de Stefan-Boltzmann [4].

$$E_n = \sigma T_s^4 \quad (2.6)$$

Em que:

$E_n$  – Poder emissivo [ $W/m^2$ ];

$\sigma$  – Constante de Stefan-Boltzmann, com um valor igual a  $5,67 \times 10^{-8}$  [ $W/(m^2K^4)$ ]

$T_s^4$  – Temperatura absoluta [K].

## **2.2. Metodologias de cálculo de cargas térmicas em programas informáticos**

Segundo Roriz (2007), as cargas térmicas podem ser divididas em três grandes grupos. O primeiro grupo são as cargas térmicas externas e estas dependem das condições climáticas. Este tipo de cargas térmicas resulta da transferência de calor e massa através da envolvente. Como segundo grupo, são apresentadas as cargas térmicas internas. Este tipo de carga térmica depende do funcionamento do edifício. São cargas térmicas que dependem do metabolismo das pessoas, iluminação e equipamentos existentes no local em análise. A carga térmica devida ao metabolismo das pessoas existe sobre a forma de latente e sensível, uma vez que é libertado vapor de água. A carga térmica sensível é a carga devida ao calor sensível, sendo este tipo de energia a que traduz um aumento de temperatura no espaço. O calor latente é o calor libertado que traduz a mudança de estado do ar, não existindo qualquer ganho térmico no espaço. Como carga térmica interna também são contabilizados os ganhos de calor através da estrutura interior do espaço climatizado. A carga térmica resultante de infiltrações de ar poderá também ser considerada como uma carga térmica interna, no caso de a infiltração ser proveniente da estrutura interior do edifício. O último grupo de tipologias de cargas térmicas é designado por outras cargas térmicas. Neste tipo de cargas estão englobados todos os ganhos de calor provenientes do funcionamento de equipamentos auxiliares como por exemplo bombas e ventiladores [5]. Para efeitos de cálculo apenas é considerado o efeito dos

ventiladores, uma vez que a energia das bombas é baixa e essa energia não traduz num aumento sensível da temperatura [5] . Também podem ser consideradas as perdas térmicas em condutas e tubagens que resultem numa carga térmica.

A importância do cálculo das cargas térmicas é:

- Dimensionar equipamentos e acessórios constituintes de um sistema de climatização;
- Cálculo do consumo energético de uma instalação existente consoante as condições de utilização do edifício, podendo ser feita uma análise anual ou em parte do ano.

Segundo Alves (2011), a maioria dos métodos existentes calculam a carga térmica em duas etapas. Na primeira são considerados os ganhos ou perdas de calor externos e internos. Numa segunda etapa são consideradas as cargas térmicas devidas a equipamentos auxiliares de climatização [6].

Na maioria dos casos o cálculo de cargas térmicas em edifícios é realizado através de programas informáticos. A *American Society of Heating and Air-Conditioning Engineers* (ASHRAE) apresenta cinco metodologias de cálculo que são ou podem ser utilizados em programas informáticos. Os cinco métodos são:

- Método do balanço térmico (HBM);
- Método da diferença de temperatura diferencial (TETD/TA);
- Método das funções de transferência (TFM);
- Método da diferença de temperatura (CLTD/SCL/CLF);
- Método das séries radiactivas (RTS).

### **2.2.1. Método do balanço térmico**

O HBM é um modelo matemático que permite o cálculo das cargas térmicas de uma forma directa sem a utilização de transformadas ou de parâmetros atribuídos arbitrariamente [7].

Este método tem por base uma abordagem matemática rigorosa, normalmente requer o uso de computadores. Muitos programas energéticos de simulação utilizam esta metodologia de cálculo de cargas térmicas. O primeiro programa a utilizar este método foi o “*National Bureau of Standards Load Determination*” (NBSLD) desenvolvido por

Tamami Kusuda, embora muitos outros programas tenham implementado este modelo matemático no seu código [7].

A teoria mais relevante deste método é que o ar interior do espaço é modelado de forma a que seja considerado totalmente misturado, isto é, o espaço tem a mesma temperatura em qualquer zona. As outras teorias referem que as superfícies do espaço (paredes, janelas, pavimentos, etc.) podem ser tratadas como ou tendo [7]:

- Temperaturas de superfície uniformes;
- Superfícies radiactivas difusas;
- Condução unidimensional através das superfícies.

Este modelo matemático pode ser dividido em quatro processos [7]:

- Balanço térmico às superfícies exteriores;
- Processo de condução térmica através das paredes;
- Balanço térmico relativamente às superfícies interiores;
- Balanço térmico relativamente ao ar.

A metodologia de cálculo apresentada através do método do balanço térmico pressupõe a definição de pelo menos uma zona térmica. O balanço térmico para esta zona envolve análises de 24 horas por dia às temperaturas interiores e exteriores às superfícies da zona, bem como ao sistema de Aquecimento, Ventilação e Ar Condicionado (AVAC). Nas cargas internas da zona térmica é necessário fazer a divisão entre cargas internas sensíveis e latentes dependendo do tipo de fonte de calor [6].

### **2.2.2. Método da diferença de temperatura diferencial**

O método TETD/TA numa primeira fase foi direccionado para ser utilizado como método de cálculo manual. Para tal, são utilizados valores tabelados para os factores de deterioração, atrasos temporais nos ganhos térmicos e de diferença de temperaturas diferencial (TETD). Estes dados são baseados nas soluções da série de Fourier para a solução da equação de condução unidimensional e não estacionária [8].

TETD/TA é um método que converte de uma forma simplificada os ganhos de calor em cargas térmicas [9]. Segundo ASHRAE (1997), com uma simples variação da carga devida a radiação o modelo não se aproxima da realidade, outro aspecto menos positivo

neste método é a escolha subjectiva na variação temporal que depende muito da experiência de quem utilizar este método [8].

### **2.2.3. Método das funções de transferência**

O método das funções de transferência (TFM) é a simplificação do método do balanço térmico [9]. Este método aplica uma série de factores de ponderação ou coeficientes condutivos da função de transferência (CTF) de várias superfícies exteriores opacas. A TFM aplica uma segunda série de coeficientes ponderativos ou coeficientes de zona da função de transferência (RTF) aos valores de ganho de calor e carga térmica de arrefecimento, para contabilizar o efeito de armazenamento térmico na conversão do ganho de calor em carga térmica [8].

Devido à facilidade de utilização dos valores de entrada e saída como também à economia de tempo de computação desta metodologia de cálculo, torna o método das funções de transferência, um dos métodos mais utilizados em programas de cálculo das cargas térmicas [9].

### **2.2.4. Método da diferença de temperatura**

O método da diferença de temperatura (CLTD / SCL/ CLF - *Cooling Load Temperature Difference / Solar Cooling Load Factor / Cooling Load Factor*) é um método de cálculo de cargas térmicas baseado no método da função de transferência (TFM). Devido à sua forma simplista e aos coeficientes multiplicativos, este método pode ser tanto utilizado manual ou computacionalmente [8]. O método tem em consideração o atraso temporal dos ganhos internos condutivos através de superfícies opacas exteriores e o atraso temporal dos ganhos de calor internos devido à radiação em carga térmica no espaço [6].

Este método apresenta três tipos de coeficientes multiplicativos. O primeiro coeficiente é a diferença de temperatura teórica (CLTD). A diferença de temperatura teórica pretende representar o efeito combinado da diferença de temperatura interior do espaço e a temperatura exterior ao espaço, variação de temperatura ao longo do dia, radiação solar e eventuais fenómenos de inércia térmica. O CLTD diz respeito a ganhos térmicos externos. O coeficiente multiplicativo que representa os ganhos térmicos por transferência através

da envolvente envidraçada é o factor solar (SCL). Para representar o facto de o ganho interno devido à radiação não se tornar imediatamente em carga térmica é apresentado o coeficiente de carga térmica (CLF). Este coeficiente serve de ajuste aos ganhos de calor internos [6].

### **2.2.5. Método das séries radiactivas**

O método das séries radiactivas (RTS – *Radiant Time Series*) é um método simplificado que deriva do método do balanço térmico. Este método veio substituir todos os anteriores métodos simplificados (TFM, CLTD/CLF e TEDTD/TA), tendo sido desenvolvido com o objectivo de ser uma metodologia rigorosa sem necessitar de iterações e que quantifique a porção que cada componente contribui para a carga térmica total. O método RTS é adequado para o cálculo das cargas térmicas de pico, mas não deve ser utilizado para simulação energética anual devido às suas suposições. Também não é aconselhável calcular as cargas térmicas manualmente, mas sim fazer uma implementação computacional [7].

A base da derivação do método do balanço térmico para as séries radiactivas é a suposição que as condições são constantes periodicamente, isto é, o ganho térmico de um componente a uma dada hora é o mesmo 24 horas antes [7].

Este método contabiliza dois tipos de atrasos na conversão de cargas térmicas, sendo eles o atraso devido à condução e à radiação. O atraso devido à condução ocorre devido aos materiais de construção terem massa e capacidade térmica, provocando um aumento de tempo até que o calor atravesse o componente. Quanto ao atraso devido à radiação, este ocorre uma vez que a radiação numa primeira fase deverá ser absorvida pelas superfícies interiores da envolvente opaca, e só depois de ser absorvida é que é transferida por convecção para o ar do espaço, tornando a radiação em carga térmica do espaço [7]. Numericamente o método contabiliza os atrasos multiplicando os ganhos de calor horários por factores de resposta. Os factores de resposta utilizados são os factores temporais radiactivos e os factores temporais condutivos. Os factores temporais radiactivos reflectem a percentagem de ganho térmico por radiação da hora anterior, que é contabilizado na carga térmica na hora corrente. Enquanto, os factores temporários condutivos reflectem a percentagem de ganho térmico por condução da hora anterior, sendo contabilizado na carga térmica na hora corrente [7].

## **2.3. Programas informáticos de simulação**

Segundo Roriz (2007), os programas informáticos de simulação têm vindo a ser utilizados desde a década de 1970, quer programas desenvolvidos por fabricantes quer por Instituições Públicas [5]. Devido à evolução tecnológica, ao aumento dos cuidados a ter com questões ambientais e as suas implicações na racionalização energética, cuidados com a saúde pública, conforto térmico e a aspectos económicos, levaram à necessidade de elaborar legislação com o intuito de reduzir o consumo energético. Com isto, cada vez mais recorre-se a programas informáticos de simulação utilizando-os como ferramentas de verificação, permitindo em muitos casos a simulação de situações reais com uma grande aproximação [5].

Os programas de simulação que calculam as cargas térmicas, na sua essência utilizam equações de transferência de calor e massa associadas a uma série de algoritmos e estruturas de dados, que permitem a obtenção de valores, consoante a introdução de características do edifício. A maioria dos programas permitem determinar o comportamento térmico do edifício, como também saber o seu consumo energético [5].

Com base em alguns programas mais utilizados, verifica-se que existem grandes diferenças no detalhe da informação desenvolvida e da forma como é fornecida. Comparando diferentes programas de simulação verifica-se que apesar de os resultados obtidos não serem contraditórios, registam-se algumas disparidades [5].

Existe um grande número de programas que possibilitam a simulação dinâmica do comportamento térmico de edifícios. Tendo esses programas como limitações geometrias complexas e respectivos sombreamentos como também a implementação de sistemas de ar condicionado menos comuns [5].

### **2.3.1. Hourly Analysis Program**

O programa Hourly Analysis Program (HAP) é um programa desenvolvido pela Carrier que funciona em ambiente Windows. Este programa é bastante intuitivo e simples, o que permite rapidamente obter resultados.

A metodologia de cálculo utilizada por este programa é o método da função de transferência (TFM) da ASHRAE [10]. Este programa permite fazer o cálculo de cargas

térmicas de aquecimento e arrefecimento a cada hora, nos doze meses do ano. Calcula também o fluxo de calor que atravessa ou é produzido por cada elemento construtivo ou equipamento. No cálculo de cargas térmicas além de fazer uma análise horária ao longo do ano, também é possível introduzir sazonalidade na simulação uma vez que permite definir vários horários de ocupação, iluminação ou de equipamentos para o mesmo espaço.

Em relação à estrutura de dados, o HAP disponibiliza uma estrutura de dados climatérica com dados de projecto referentes a mais de 800 cidades de vários países e para realizar simulações disponibiliza dados referentes a mais de 500 cidades de vários países. A extensão dos ficheiros climáticos do HAP são: TMY, IWEW, CWEW, TMY e TRY. Embora também seja possível importar dados com as seguintes extensões: EPW, NOAA, TMY3 e IEWC2 [10]. Além da estrutura de dados climática o HAP também disponibiliza estruturas de dados de vidros, materiais de construção, e de ganhos térmicos sensíveis e latentes de ocupação.

Esta aplicação disponibiliza o manual do programa em formato PDF e um vasto menu de ajuda. Quanto aos relatórios, estes são exportados no formato PDF e permitem escolher o tipo de análise que queremos fazer. Na parte dos relatórios existe também a possibilidade de gerar uma carta Psicrométrica. A Figura 2.3 apresenta o menu inicial do programa HAP.

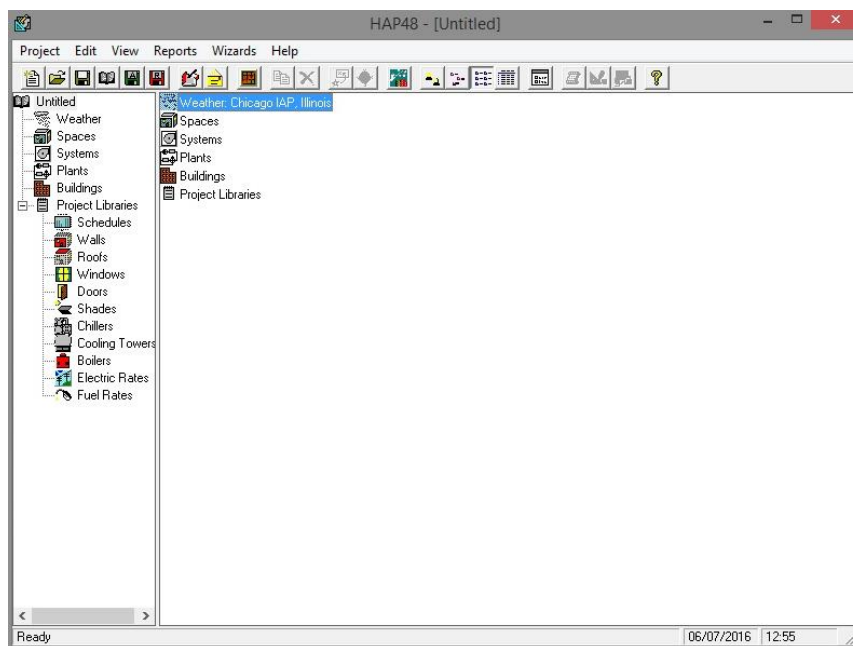


Figura 2.3. Janela inicial do HAP.

### 2.3.2. EnergyPlus

O *EnergyPlus* foi oficialmente apresentado em 1996, sendo um dos programas mais utilizados actualmente quer no meio académico que no meio empresarial. Segundo Alves (2011) a aplicação tem estado em constante actualização e resulta da parceria entre o Laboratório Nacional Lawrence Berkeley, Universidade de Ilinoís, Laboratório de pesquisa de Engenharia de Construção das Forças Armadas dos Estados Unidos da América, e o Escritório de Tecnologia das Edificações [6]. Esta aplicação resulta da combinação de dois programas de simulação (DOE-2 e BLAST), gerando uma aplicação com um interface mais acessível e uma arquitectura informática que permite actualizações e interacção com outras ferramentas [5]. Este programa permite o cálculo de cargas térmicas e consumos energéticos numa base horária e em múltiplas zonas. O método de cálculo das cargas térmicas baseia-se no método do balanço térmico, sendo considerado que a temperatura do ar interior é uniforme [11].

O *EnergyPlus* é um conjunto de sistemas modulares que funcionam em ambiente Windows, Linux e Mac. É uma aplicação licenciada de código aberto, permitindo a criação de ferramentas de apoio, ou até mesmo módulos de cálculo que não estejam disponíveis na versão base do programa [11]. A Figura 2.4 representa a janela “Início” do programa *EnergyPlus*.

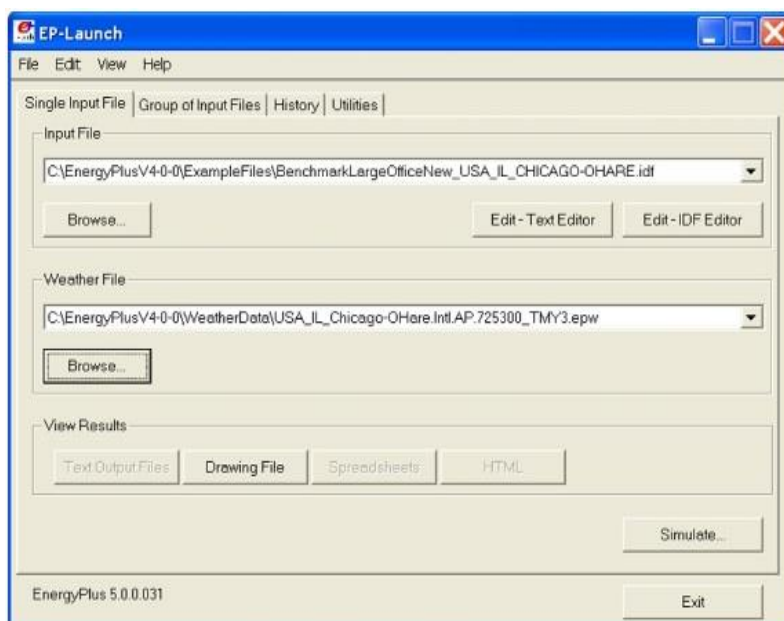


Figura 2.4. Energy Plus – Janela “Início” [11].



### 2.3.3. DOE-2

A aplicação DOE-2 foi desenvolvida por James *J.Hirsch & Associates* (JH), em colaboração com *Lawrence Berkeley National Laboratory* (LBNL) [12]. Segundo Alves (2011) o DOE-2 é composto por dois elementos, o primeiro é um processador de dados do edifício denominado *Building Description Language* (BDL) e um conjunto de subaplicações que utilizam os valores introduzidos pelo utilizador na BDL para a realização da simulação energética horária [6]. Os quatro subprogramas chamam-se *Loads*, *Systems*, *Plants* e *Economics*. O *Loads* calcula as cargas térmicas para cada espaço do edifício, o subprograma *Systems* simula os ganhos internos causados pelo equipamento interior, ventilação [12].

Para além do que foi referido anteriormente, o DOE-2 disponibiliza ainda:

- Subprograma de impressão de resultados;
- Processador de dados climatéricos, sendo possível introduzir informação horária do clima de cada local;
- Estrutura de dados de materiais e elementos construtivos de edifícios, como por exemplo: paredes, tectos, janela, etc.;
- Possibilidade de fazer alterações na estrutura de dados.

Esta aplicação é de livre acesso, sendo a sua versão mais recente a DOE-2.3 versão 49m, lançada em 14/03/2016, que funciona unicamente em ambiente Windows [12].

### 2.3.4. Integrated Environmental Solutions

Segundo Roriz (2007), o conjunto de aplicações IES é um programa que possui interfaces de entrada e saída de um conjunto de ferramentas, que permitem a simulação de cargas térmicas, consumos energéticos, aspectos luminotécnicos, controlo, utilização de elevadores, entre outros parâmetros [5].

Esta aplicação desenvolve o modelo em ambiente de desenho assistido por computador (DAC), dando a possibilidade de criar novos elementos, como por exemplo: sistemas, equipamentos principais, materiais, etc. No que diz respeito à exportação de dados a

aplicação disponibiliza ao utilizador a possibilidade de cruzar as variáveis que pretender [5].

### 2.3.5. Trasient System Simulation Program

Segundo Alves (2011), o programa *Trasient System Simulation Program* (TRNSYS) é desenvolvido pelo Laboratório de Energia da Universidade de Wisconsin, tendo sido disponibilizado comercialmente desde 1975. Esta aplicação tem como finalidade simular o desempenho transiente de um sistema de energia térmica. O funcionamento do programa baseia-se numa estrutura modular, permitindo ao utilizador adicionar modelos matemáticos, reduzindo a complexidade da simulação [6].

O programa TRNSYS está dividido em duas partes distintas. A primeira parte é um motor, denominado *Kernel*, que lê e processa o ficheiro de entrada, resolve o modelo iterativamente, determina convergências e exporta os resultados da simulação do modelo [13]. A segunda parte é uma vasta estrutura de dados que contempla desde ficheiros de dados climatéricos até dados de equipamentos de climatização [13].

De acordo com Alves (2011), a base do TRNSYS é de simples compreensão e como vantagens apresenta a sua flexibilidade, modularidade, ser uma aplicação de código aberto e ter uma vasta estrutura de dados. Como desvantagens destaca-se a complexidade para novos utilizadores, dificuldade em identificar erros do modelo e o facto de a aplicação ser comercial [6]. A Figura 2.5 representa o módulo de simulação (*Simulation Studio*) do programa TRNSYS.

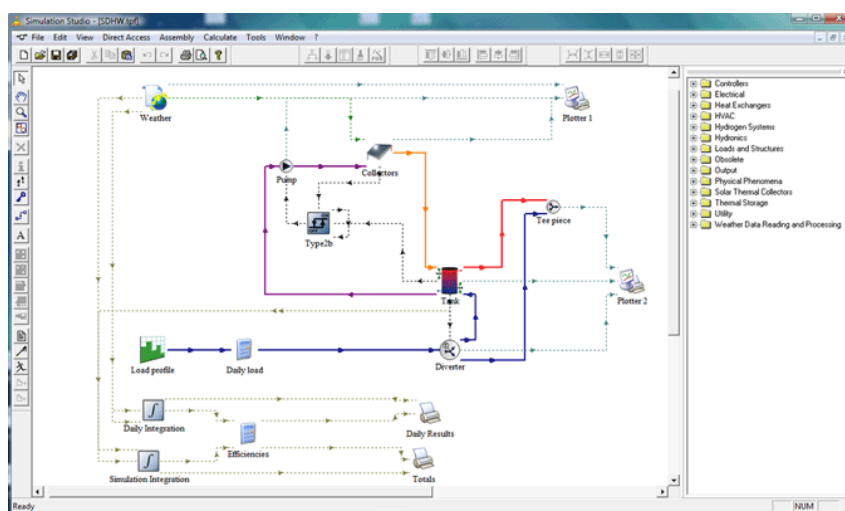


Figura 2.5. Simulation Studio TRNSYS [13].

### 2.3.6. Trane Air Conditioning Economics

O *Trane Air Conditioning Economics* (TRACE 700) é um programa que funciona em ambiente *Windows* comercializado pela *Trane*, que é um fabricante de equipamentos de climatização. Esta aplicação de código fechado permite a simulação energética de edifícios e analisa as cargas térmicas e sistemas de climatização de um ponto de vista energético e económico [6].

A aplicação é dotada de uma estrutura de dados de materiais de construção, elementos construtivos, cargas internas, ficheiros climáticos, horários de utilização e de equipamentos de climatização [14]. Permitindo definir e adicionar à estrutura de dados componentes personalizados. Para tal, é disponibilizado um editor que serve para construir/ editar a estrutura de dados. Quanto aos ficheiros climáticos, a aplicação permite a importação de ficheiros com a extensão TMY, TMY2, TMY3, TRY, WY2, CEC, CTZ, CWC e IWECC [15]. Permite também importar e exportar ficheiros DAC com a extensão gbXML e exporta os dados da simulação nos seguintes formatos: PDF, RTF, Word e Excel [14].

O TRACE 700 disponibiliza oito metodologias de cálculo de cargas térmicas, sendo uma delas o balanço térmico baseado no RTS [14]. A Figura 2.6 representa o *Project Navigator* do programa TRACE 700.

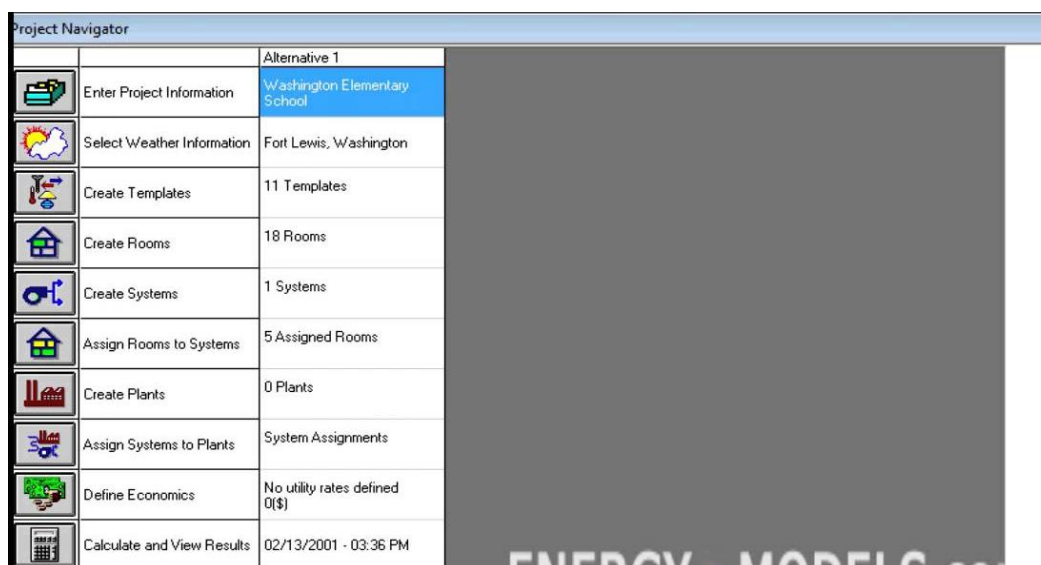


Figura 2.6. *Project Navigator*, TRACE 700.

## **2.4. Soluções construtivas**

A Informação Técnica de Edifícios 50 (ITE 50) do Laboratório Nacional de Engenharia Civil (LNEC) apresenta uma panóplia de soluções construtivas, sendo elas paredes de fachada simples e duplas, pavimentos sobre espaços exteriores, cobertura horizontais e inclinadas e vãos envidraçados.

### **2.4.1. Paredes de fachada**

As paredes de fachada simples abrangem várias soluções. Essas soluções são divididas em dois grupos. Sendo esses grupos a solução construtiva da parede e soluções de isolamento térmico. Na solução construtiva da parede são consideradas as paredes de alvenaria simples e a parede moldada de betão simples ou armado, de inertes correntes com 0,10 m a 0,20 m de espessura. Na solução de parede de alvenaria simples pode ser feita uma diferenciação consoante as características dos materiais utilizados [3]:

- Alvenaria de tijolo furado de barro vermelho;
- Alvenaria de blocos de inertes correntes;
- Alvenaria de blocos de betão leve com inertes de argila expandida;
- Alvenaria de pedra.

Das soluções apresentadas anteriormente, pode-se considerar duas alternativas de revestimento superficial, sendo eles o revestimento aderente em ambas as faces da parede e o revestimento independente numa das faces da parede. As soluções de isolamento térmico compreendem paredes com isolamento térmico aplicado no exterior e no interior [3].

A outra tipologia de parede de fachada é a parede dupla que abrange a solução construtiva da parede e a solução de isolamento térmico. A solução construtiva apresenta as seguintes tipologias [3]:

- Parede dupla com panos de alvenaria;
- Parede dupla com um pano de betão de inertes e outro de alvenaria.

Na solução de isolamento térmico as tipologias apresentadas no ITE 50 são:

- Isolante térmico preenchendo totalmente o espaço intermédio entre os panos da parede;
- Isolante térmico preenchendo parcialmente aquele espaço intermédio.

## 2.4.2. Pavimentos

O pavimento é a superfície plana que permite caminhar comodamente na sua área, como também é resistente ao desgaste. A escolha do pavimento, varia consoante o tipo de ocupação do local [16].

A constituição genérica de um pavimento de um piso está representada na Figura 2.7. Sendo composto por um elemento estrutural, impermeabilização, isolamento térmico, betonilha, camada de assentamento e revestimento [17].

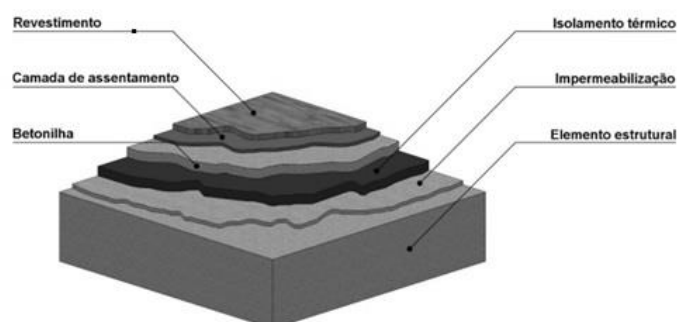


Figura 2.7. Camadas genéricas constituintes de um pavimento [17].

No caso de o pavimento ser de um piso térreo, existe a necessidade de impermeabilização, por isso o esquema da tipologia de construção é de acordo com a Figura 2.8 [16].

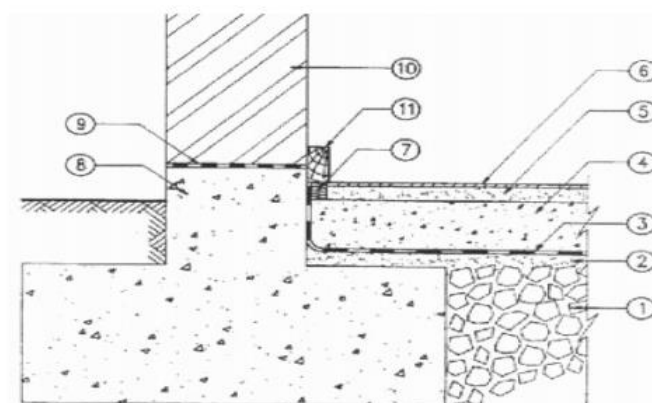


Figura 2.8. Impermeabilização de pavimento térreo [16].

Legenda:

- |                                       |                                                          |
|---------------------------------------|----------------------------------------------------------|
| 1 – Enrocamento;                      | 7 – Junta de dessolidarização com material compressível; |
| 2 – Camada de regularização em areia; | 8 – Maciço de fundação de parede;                        |
| 3 – Folha de polietileno;             | 9 – Camada de impermeabilização;                         |
| 4 – Massame de betão;                 | 10 – Parede exterior;                                    |
| 5 – Betonilha de regularização;       | 11 – Rodapé.                                             |
| 6 – Revestimento de piso;             |                                                          |

### 2.4.3. Coberturas

O elemento construtivo cobertura é dividido em duas tipologias construtivas. A primeira tipologia é a cobertura horizontal (em terraço), podendo ser acessível ou não. Uma cobertura denomina-se em terraço quando os materiais que a constituem estão dispostos horizontalmente ou próximo dessa posição [17]. As camadas constituintes de uma cobertura em terraço são [18]:

- Suporte resistente (contínuo ou pontual) – Elemento de suporte da cobertura, pode ser uma laje contínua ou um conjunto de elementos pontuais;
- Camada de forma – Camada que irá garantir a pendente da cobertura. Quando executada com betões leves deve ser regularizada superficialmente com uma betonilha;
- Camada de difusão – Em zonas de forte higrometria (muita produção de vapor de água) é utilizada para distribuir a pressão de vapor;
- Barreira para vapor – Impede a passagem do vapor de água;
- Isolamento térmico – Material com grande resistência térmica, que reduz as trocas térmicas entre o interior e o exterior;
- Camada de independência – Garante a separação entre o isolamento térmico e a impermeabilização;
- Revestimento de impermeabilização – Material impermeável, como por exemplo uma tela asfáltica;
- Camada de dessolidarização – Permite o funcionamento independente da impermeabilização e da camada de protecção, ajudando à preservação da impermeabilização;

- Camada de protecção – Para protecção da impermeabilização, mas também para acabamento final da cobertura.

A segunda tipologia de cobertura é a cobertura inclinada, com ou sem desvão. Existem várias configurações para este tipo de coberturas, que dependem essencialmente do local e da posição do isolamento. Os isolamentos térmicos considerados neste tipo de coberturas são: placas de aglomerado negro de cortiça, poliestireno expandido moldado, poliestireno expandido extrudido, espuma rígida de poliuretano, mantas ou placas de lã mineral ou por camadas de grânulos de argila expandida a granel [19].

Quando o espaço por baixo da cobertura não é utilizado para habitação ou lazer, isto é desvão não habitável, é melhor utilizar a camada de isolamento térmico sobre a esteira horizontal. Na Figura 2.9 estão representados três tipos de coberturas inclinadas com desvão não habitável.

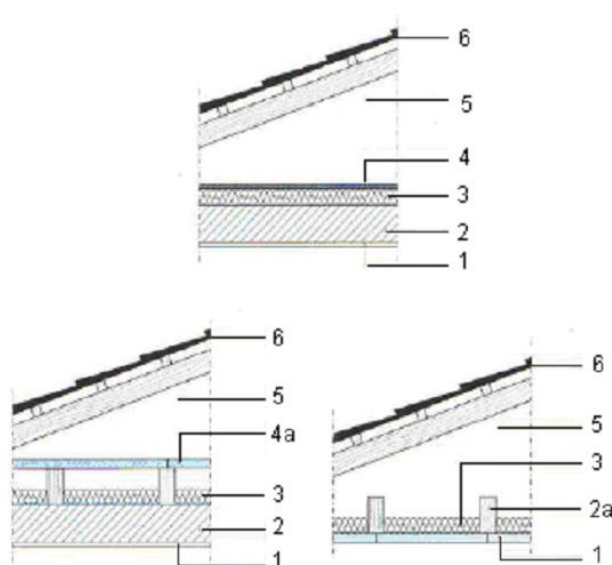


Figura 2.9. Cobertura inclinada com desvão não habitável – Isolamento térmico na esteira horizontal [19].

Legenda:

- |                                       |                                               |
|---------------------------------------|-----------------------------------------------|
| 1 – Revestimento de tecto;            | 4 – Protecção superior do isolamento térmico; |
| 2 – Laje de esteira;                  | 4a – Revestimento de piso;                    |
| 2a – Estrutura de madeira da esteira; | 5 – Desvão;                                   |
| 3 – Isolamento térmico;               | 6 – Revestimento da cobertura.                |

Na Figura 2.10 verifica-se que o isolamento térmico está na vertente inclinada. As soluções com isolamento térmico aplicado nas vertentes das coberturas inclinadas devem apenas ser utilizadas nos casos em que o desvão é habitável.

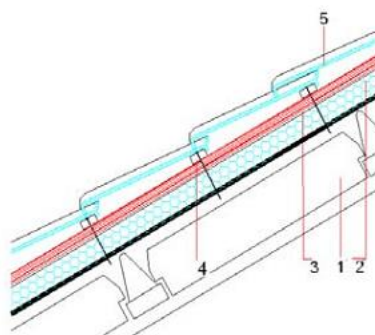


Figura 2.10. Cobertura inclinada com desvão habitável – Isolamento térmico nas vertentes [19].

Legenda:

- |                         |            |
|-------------------------|------------|
| 1 – Laje;               | 4 – Ripa;  |
| 2 – Isolamento térmico; | 5 – Telha. |
| 3 – Subtelha;           |            |

#### 2.4.4. Vão envidraçado

Segundo Mendes (2011), um vão envidraçado é uma abertura na parte opaca de um edifício, destinada a receber um caixilho. Os vãos envidraçados são compostos por duas partes: o aro e a folha. O aro é uma peça com as dimensões do vão e é colocada à volta deste. No caso de uma porta, o aro é constituído por dois montantes e uma travessa, formando um U invertido. As folhas são elementos que se ajustam aos aros e têm várias formas de movimentos, tais como de abertura e fecho total ou parcial. São constituídas por caixilho, vidros e eventualmente por painéis opacos [20].

Segundo a norma BS EN ISO 10077-1:2006, as janelas podem ser classificadas como janelas simples, duplas e de dupla folha no mesmo aro. Podendo qualquer tipo ter dispositivo de oclusão [21]. Na Figura 2.11 é representado a constituição de um vão envidraçado, nomeadamente de uma janela.

No caso das portas, estas podem ser consideradas vãos envidraçados ou não, dependendo se este elemento possuir vidro. A norma BS EN ISO 10077-1:2006, refere que a metodologia de cálculo do coeficiente global de transmissão de calor é rigorosamente



igual, tendo só de ser definido o valor da área de vidro como sendo igual a zero [21]. A Figura 2.12 representa os constituintes do vão envidraçado porta.

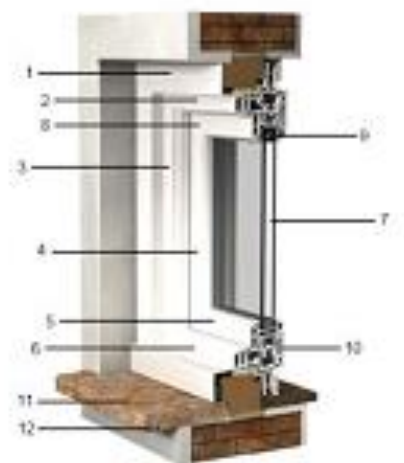


Figura 2.11. Definição dos elementos da janela [20].

Legenda:

- |                                |                                       |
|--------------------------------|---------------------------------------|
| 1 – Pré-Aro                    | 7 – Sistema envidraçado               |
| 2 – Aro (fixo)                 | 8 – Folha (“caixilho” ou “aro móvel”) |
| 3 – Couceira                   | 9 – Perfil intercalar (espaçador)     |
| 4 – Couceira da folha          | 10 – Septos                           |
| 5 – Travessa inferior da folha | 11 – Parapeito                        |
| 6 – Tábua de peito             | 12 – Goteira                          |



Figura 2.12. Definição dos elementos da porta [20].

Legenda:

1 – Vidro

4 – Almofada

2 – Folha

5 – Travessa superior

3 – Montante

6 – Aro

#### **2.4.5. Caixilharia**

A caixilharia é o componente do vão envidraçado que pretende garantir a permeabilidade ao ar e à água. Escolher a caixilharia adequada pode traduzir na diminuição do consumo energético do edifício, através da redução das perdas térmicas pelos vãos envidraçados. Com o desenvolvimento de caixilharias estanques consegue-se de uma forma mais eficaz controlar as trocas de calor entre o espaço interior e exterior [3]. Segundo Mendes (2011), a classificação das caixilharias pode ser realizada de acordo com a sua função (janelas, portas e outras), material utilizado (madeira, alumínio, PVC, etc.) ou até mesmo a forma de abertura da folha (batente, basculante, pivotante, entre outras) [20].

Este componente costuma ocupar cerca de 20% a 30% da área do vão envidraçado. Embora seja uma porção diminuta do vão envidraçado, o desempenho térmico das caixilharias contribui significativamente no coeficiente global de transmissão de calor do vão [3].

#### **2.4.6. Vidro**

O vidro é o componente transparente ou translúcido de um vão envidraçado. A área de vidro ocupa cerca de 70% a 80% de um vão envidraçado [3]. Os vidros podem ser classificados em simples, duplos ou até mesmo em vidros triplos. Um vão envidraçado com vidro simples é composto por uma única folha de vidro. No caso de ser um vão envidraçado com vidro duplo, o vão é composto por duas folhas de vidro simples nas extremidades. Entre os vidros simples está presente o espaçador. O espaçador pode ser composto por ar ou até mesmo por gases (árgon, xénon, cripton), contribuindo na diminuição das perdas térmicas do edifício [20]. Um vão envidraçado com vidro triplo é composto por três folhas de vidro separadas por dois espaçadores. A nomenclatura utilizada no elemento construtivo denominado vidro é representado na figura 2.13.

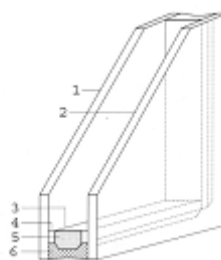


Figura 2.13. Representação de um vidro de isolamento térmico (vidro duplo) [20].

Legenda:

1 – Exterior

2 – Interior

3 – Perfil intercalar (Espaçador)

4 – Vedante

5 – Material desidratante

6 – Vedante



### 3. Formulação Matemática

Neste capítulo é apresentada a metodologia de cálculo dos coeficientes globais de transmissão de calor essenciais para calcular as cargas térmicas. A psicrometria é abordada neste capítulo com o intuito de obter as entalpias do ar exterior e interior, a partir das condições de temperatura de bulbo seco e de humidade relativa. Sendo estas entalpias relevantes para o cálculo das cargas térmicas devidas a infiltrações de ar.

#### 3.1. Coeficiente global de transmissão de calor da envolvente opaca.

A transmissão de calor por condução e convecção são causas decisivas nas perdas de calor por transmissão através da envolvente de um edifício.

Uma vez que cada material construtivo apresenta uma determinada resistência, adicionando essa resistência às resistências convectivas superficiais (exteriores e interiores), resulta o coeficiente global de transmissão de calor, U. Este coeficiente define a maior ou menor dificuldade que um material oferece à transmissão de calor [2]. O valor, U, de elementos construtivos como por exemplo, pavimentos, tectos, coberturas e paredes é calculado de acordo com a equação (3.1).

$$U = \frac{1}{R_{si} + \sum \frac{d}{\lambda} + R_{se}} \quad (3.1)$$

Em que:

$R_{se}$  – Resistência térmica exterior  $[(m^2 \cdot K)/W]$

$\lambda$  – Condutibilidade térmica do material  $[W/(m \cdot K)]$ ;

$d$  – Espessura do material  $[m]$ ;

$R_{si}$  – Resistência térmica interior  $[(m^2 \cdot K)/W]$ .

O valor do coeficiente global de transmissão de calor varia consoante os materiais e espessuras dos constituintes do elemento construtivo. Mas é também dependente das resistências convectivas internas e externas. Essas resistências em teoria podem-se considerar constantes, embora variem com a velocidade do escoamento. Na Tabela 3.1 são apresentados os valores das resistências convectivas para o sentido do fluxo de calor

horizontal e vertical. No caso vertical também são apresentados os sentidos ascendentes e descendentes do fluxo de calor.

Tabela 3.1. Resistências Térmicas Superficiais [3].

Sentido do fluxo de calor	Resistência térmica superficial [(m <sup>2</sup> .°C)/W]	
	Exterior R <sub>se</sub>	Interior R <sub>si</sub>
Horizontal	0,04	0,13
Vertical		
Ascendente	0,04	0,10
descendente	0,04	0,17

## 3.2. Vãos envidraçados

Neste subcapítulo são definidas as metodologias de cálculo do coeficiente global de transmissão de calor de vãos envidraçados. Os vãos envidraçados em estudo são: janelas simples, janelas duplas, janelas acopladas, janelas com dispositivo de oclusão fechado e portas. Antes de apresentar a metodologia de cálculo dos coeficientes globais de transmissão de calor serão definidas as várias áreas constituintes de um vão envidraçado.

### 3.2.1. Definição Geométrica do envidraçado

A área do vidro, A<sub>g</sub>, ou do painel opaco, A<sub>p</sub>, de uma janela ou portas é a menor das áreas visíveis de ambos os lados. Sendo que as sobreposições de juntas devem ser ignoradas [21].

O perímetro total do vidro, I<sub>g</sub>, (ou do painel opaco, I<sub>p</sub>) é a soma dos perímetros visíveis dos painéis de vidro (ou painéis opacos) de uma janela ou porta. No caso de os perímetros de ambos os lados serem diferentes, deverá ser utilizado o de maior valor [21]. Na Figura 3.1 estão identificados a área do vidro e o seu perímetro.

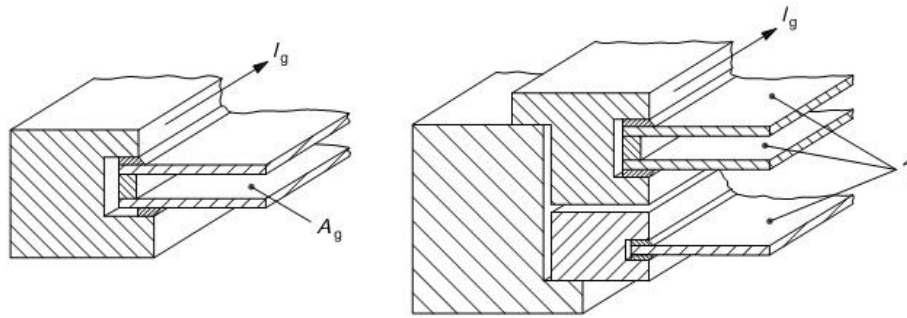


Figura 3.1. Perímetro e área do vidro [21].

A área do perfil da caixilharia,  $A_f$ , é o maior valor entre a área projectada interior ( $A_{f,i}$ ) e a exterior ( $A_{f,e}$ ). A área da caixilharia inclui a área do aro e da folha [21].

$$A_f = \max(A_{f,i}, A_{f,e}) \quad (3.2)$$

A área da janela é a soma entre a área do perfil ( $A_f$ ) e a área do vidro ( $A_g$ ) [21]. A equação (3.3) permite o cálculo da área da janela.

$$A_w = A_f + A_g \quad (3.3)$$

A área de desenvolvimento interno e externo do caixilho são calculadas a partir da equação (3.4) e da equação (3.5) respectivamente.

$$A_{f,di} = A_1 + A_2 + A_3 + A_4 \quad (3.4)$$

$$A_{f,de} = A_5 + A_6 + A_7 + A_8 \quad (3.5)$$

A Figura 3.2 representa as diferentes áreas de um vão envidraçado.

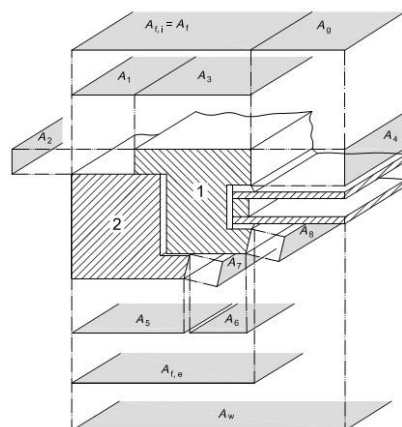


Figura 3.2. Representação das diversas áreas de um vão envidraçado [21].

As áreas de desenvolvimento interno e externo do caixilho são definidas na figura 3.3.

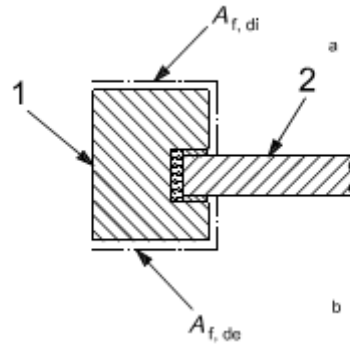


Figura 3.3. Área de desenvolvimento do caixilho [21].

### 3.2.2. Vidro simples e duplo

O coeficiente global de transmissão de calor,  $U_g$ , de um vidro simples é calculado usando a equação (3.6) [21].

$$U_g = \frac{1}{R_{se} + \sum \frac{d_v}{\lambda_v} + R_{si}} \quad (3.6)$$

Em que:

$R_{se}$  – Resistência térmica exterior  $[(m^2 \cdot K) / W]$ ;

$\lambda_v$  – Condutibilidade térmica do vidro  $[W/(m \cdot K)]$ ;

$d_v$  – Espessura do vidro  $[m]$ ;

$R_{si}$  – Resistência térmica interior  $[(m^2 \cdot K) / W]$ .

Para o vidro duplo, o coeficiente global de transmissão térmica,  $U_g$ , é calculado através da equação (3.7) [21].

$$U_g = \frac{1}{R_{se} + \sum \frac{d_v}{\lambda_v} + \sum R_s + R_{si}} \quad (3.7)$$

Em que:

$R_{se}$  – Resistência térmica exterior  $[(m^2 \cdot K) / W]$ ;

$\lambda_v$  – Condutibilidade térmica do vidro  $[W/(m \cdot K)]$ ;

$d_v$  – Espessura do vidro  $[m]$ ;

$R_s$  – Resistência térmica do espaço entre o vidro interior e exterior  $[(m^2 \cdot K) / W]$ ;

$R_{si}$  – Resistência térmica interior  $[(m^2 \cdot K) / W]$ .



Os valores das resistências térmicas superficiais internas e externas, utilizados para o cálculo do coeficiente global de transmissão de calor dos vidros simples e duplos, são os da Tabela 3.1. A Tabela 3.2 detém os valores das resistências térmicas de espaços de ar não ventilados de janelas duplas ou acopladas.

Tabela 3.2. Resistências térmicas de espaços de ar não ventilados para janelas duplas ou acopladas [21].

Resistência Térmica ( $R_s$ ) [ $(m^2 \cdot ^\circ C)/W$ ]					
Espessura do espaço de ar	Um lado revestido com uma emissividade normal				Ambos os lados não são revestidos
	0,1	0,2	0,4	0,8	
mm					
6	0,211	0,191	0,163	0,132	0,127
9	0,299	0,259	0,211	0,162	0,154
12	0,377	0,316	0,247	0,182	0,173
15	0,447	0,364	0,276	0,197	0,186
50	0,406	0,336	0,26	0,189	0,179

### 3.2.3. Perfis de caixilharia

Para determinação do coeficiente global de transmissão de calor da caixilharia a norma BS EN ISO 10077-2:2012, estabelece que o envidraçado do vão seja substituído por um painel de um material de baixa condutibilidade ( $0,035 \text{ W/m.K}$ ) [22], respeitando os espaçamentos definidos na Figura 3.4.

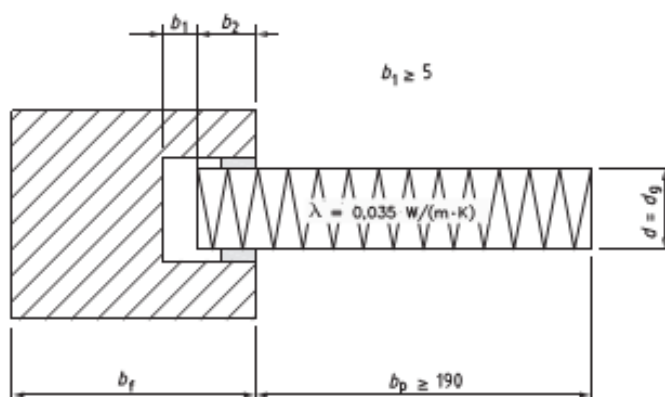


Figura 3.4. Caixilharia com painel opaco [22].

Considerando as premissas anteriores, o coeficiente global de transmissão de calor de um caixilho,  $U_f$ , é calculado de acordo com a equação (3.8) [22].

$$U_f = \frac{L_f^{2D} - U_p b_p}{b_f} \quad (3.8)$$

Em que:

$L_f^{2D}$  – Transmissão de calor bidimensional através da secção da caixilharia [W/(m.K)];

$U_p$  – Coeficiente global de transmissão de calor do painel opaco [W/(m<sup>2</sup>.K)];

$b_p$  – Largura visível do painel [m];

$b_f$  – Largura em projecção da caixilharia [m].

### 3.2.4. Janela simples

Na Figura 3.5 é feita a representação esquemática da janela simples. A janela simples tem 3 áreas distintas. A primeira área é o caixilho fixo, a segunda é o caixilho móvel e por último é o vidro. O vidro pode ser simples, duplo ou até mesmo triplo.

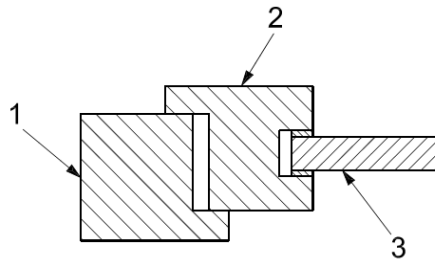


Figura 3.5. Representação esquemática de uma janela simples [21].

Legenda:

1 – Caixilho fixo;

2 – Caixilho móvel;

3 – Vidro simples ou múltiplo.

O coeficiente global de transmissão de calor de uma janela simples,  $U_w$ , deve ser calculado utilizando a equação (3.9) [21].

$$U_w = \frac{\sum A_g U_g + \sum A_f U_f + \sum I_g \Psi_g}{\sum A_g + \sum A_f} \quad (3.9)$$

Em que:

$A_g$  – Área de vidro [ $m^2$ ];

$U_g$  – Coeficiente global de transmissão de calor do vidro [ $W/(m^2.K)$ ];

$A_f$  – Área do caixilho [ $m^2$ ];

$U_f$  – Coeficiente global de transmissão de calor do caixilho [ $W/(m^2.K)$ ];

$I_g$  – Perímetro do vidro [m];

$\Psi_g$  – Coeficiente de transmissão térmica linear do vidro [ $W/(m.K)$ ].

No caso do vidro simples o último termo do numerador da equação (3.9) deve ser igual a zero, uma vez que qualquer correcção deve ser negligenciada [21].

Quando existem em simultâneo, painéis envidraçados e opacos,  $U_w$ , é calculado segundo a equação (3.10) [21]:

$$U_w = \frac{\sum A_g U_g + \sum A_p U_p + \sum A_f U_f + \sum I_g \Psi_g + \sum I_p \Psi_p}{\sum A_g + \sum A_p + \sum A_f} \quad (3.10)$$

Em que:

$A_g$  – Área de vidro [ $m^2$ ];

$U_g$  – Coeficiente global de transmissão de calor do vidro [ $W/(m^2.K)$ ];

$A_p$  – Área do painel opaco [ $m^2$ ];

$U_p$  – Coeficiente global de transmissão de calor do painel opaco [ $W/(m^2.K)$ ];

$A_f$  – Área do caixilho [ $m^2$ ];

$U_f$  – Coeficiente global de transmissão de calor do caixilho [ $W/(m^2.K)$ ];

$I_g$  – Perímetro do vidro [m];

$\Psi_g$  – Coeficiente de transmissão térmica linear do vidro [ $W/(m.K)$ ];

$I_p$  – Perímetro do painel opaco [m];

$\Psi_p$  – Coeficiente de transmissão térmica linear do painel opaco [ $W/(m.K)$ ].

O coeficiente de transmissão térmica linear do painel opaco,  $\Psi_p$ , poderá ser igual a zero, se [21]:

- Os revestimentos internos e externos do painel forem de um material com uma condutividade térmica a 0,5 W/(m.K);
- A condutividade térmica de qualquer material de ligação nas extremidades for menor que 0,5 W/(m.K).

### 3.2.5. Janela dupla

A tipologia de janela dupla, consiste na junção de duas janelas simples separadas por um perfil intercalar [20]. Na Figura 3.6 está representada esquematicamente uma janela dupla.

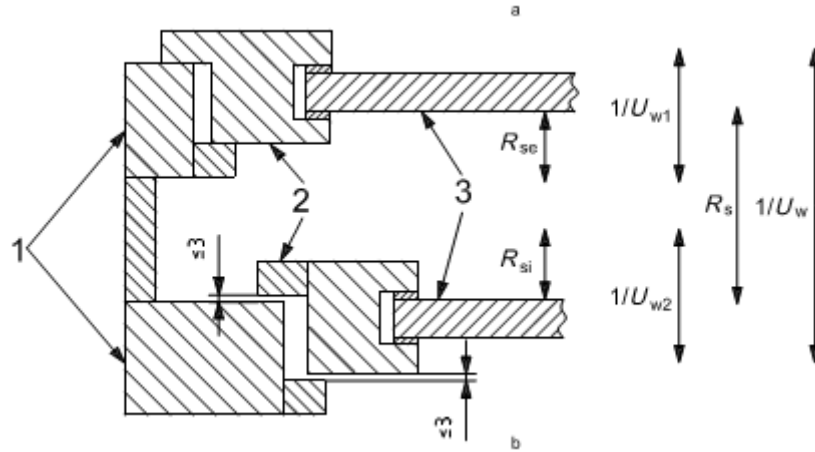


Figura 3.6. Representação esquemática de uma janela dupla [21].

Se os espaçamentos representados na Figura 3.6, excederem os 3 mm e não forem tomadas medidas para prevenir a troca de ar excessiva com o ar exterior, a metodologia de cálculo apresentada para o cálculo do coeficiente global de transmissão de calor para janelas duplas não é válido [21].

O coeficiente global de transmissão de calor de uma janela dupla é calculado de acordo com a equação (3.11) [21].

$$U_w = \frac{1}{\frac{1}{U_{w1}} - R_{si} + R_s - R_{se} + \frac{1}{U_{w2}}} \quad (3.11)$$

Em que:

$U_{w1}$ ,  $U_{w2}$  – Coeficientes globais de transmissão de calor exterior e interior da janela, respectivamente, calculados a partir da equação (3.9) [ $W/(m^2.K)$ ];

$R_{si}$  – Resistência térmica da superfície interna da janela externa, quando utilizada sozinha [ $(m^2.K)/W$ ];

$R_{se}$  – Resistência térmica da superfície externa da janela externa, quando utilizada sozinha [ $(m^2.K)/W$ ];

$R_s$  – Resistência térmica do espaço entre as duas janelas [ $(m^2.K)/W$ ].

### 3.2.6. Janela acoplada

O coeficiente global de transmissão de calor,  $U_w$ , de um sistema com caixilho fixo e dois caixilhos móveis devem ser calculados de acordo com a equação (3.9) ou pela equação (3.10) [21]. Para determinar o coeficiente global de transmissão de calor,  $U_g$ , relativamente à combinação dos dois vidros utiliza-se a equação (3.12) [21]:

$$U_g = \frac{1}{\frac{1}{U_{g1}} - R_{si} + R_s - R_{se} + \frac{1}{U_{g2}}} \quad (3.12)$$

Em que:

$U_{g1}, U_{g2}$  – Coeficientes globais de transmissão de calor do vidro interior e exterior respectivamente [ $W/(m^2.K)$ ];

$R_{si}$  – Resistência térmica da superfície interna da janela, quando utilizada sozinha [ $(m^2.K)/W$ ];

$R_{se}$  – Resistência térmica da superfície externa da janela, quando utilizada sozinha, [ $(m^2.K)/W$ ];

$R_s$  – Resistência térmica do espaço entre os dois vidros [ $(m^2.K)/W$ ].

Se os espaçamentos representados na Figura 3.7, excederem os 3 mm e não forem tomadas medidas para prevenir a troca de ar excessiva com o ar exterior, a metodologia de cálculo apresentada para o cálculo do coeficiente global de transmissão de calor para janelas acopladas não é válido [21].

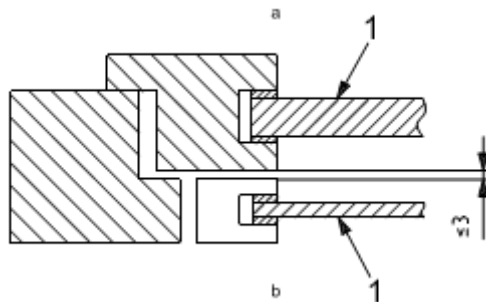


Figura 3.7. Representação esquemática de uma janela acoplada [21].

### 3.2.7. Janela com dispositivo de oclusão fechado

No caso de as janelas serem dotadas de um dispositivo de oclusão fechado, independentemente de serem janelas simples, duplas ou acopladas, o cálculo do coeficiente global de transmissão de calor é realizado através da equação (3.13) [21].

$$U_{ws} = \frac{1}{\frac{1}{U_w} + \Delta R} \quad (3.13)$$

Em que:

$U_w$  – Coeficiente global de transmissão de calor da janela [ $W/(m^2.K)$ ];

$\Delta R$  – Resistência térmica adicional devido à camada de ar entre o dispositivo de oclusão e a janela [ $(m^2.K/W)$ ].

A Figura 3.8 representa esquematicamente uma janela dotada com dispositivo de oclusão fechado.

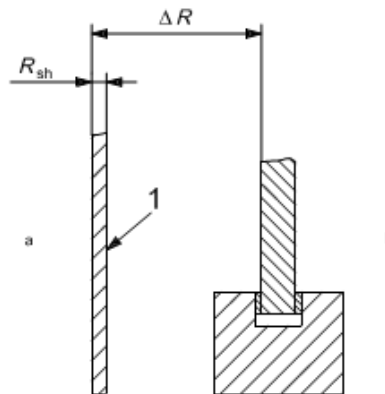


Figura 3.8. Janela com dispositivo de oclusão fechado [21].

### 3.2.8. Portas

No caso do elemento construtivo porta, o cálculo do coeficiente global de transmissão de calor de uma porta totalmente envidraçada,  $U_D$ , é calculado de acordo com a equação (3.9). Esta equação calcula o coeficiente global de transmissão de calor de janelas simples [21]. Se a porta envidraçada for dotada de painéis opacos, o cálculo do coeficiente de transmissão de calor da porta com painéis opacos é calculado através da equação (3.10), equação que calcula o coeficiente global de transmissão de calor de janelas simples com painéis opacos [21].

Caso a porta não seja um vão envidraçado, o cálculo do coeficiente global de transmissão de calor é efectuado conforme foi descrito neste subcapítulo usando a equação anteriormente descrita no subcapítulo 3.2.4. A única premissa é que a área do vidro é nula.

### 3.3. Cargas térmicas

As cargas térmicas têm uma grande variedade de origens. Podem ser ganhos de calor devido à envolvente externa ou interna, ocupação do espaço, equipamentos eléctricos, iluminação ou infiltrações de ar [9]. De seguida, será abordada a metodologia para o cálculo de cada uma destas cargas térmicas de arrefecimento (quando existe necessidade de arrefecimento do espaço) ou de aquecimento (quando existe necessidade de aquecimento do espaço).

#### 3.3.1. Ganho de calor condutivo através da envolvente exterior

Os ganhos de calor através da envolvente exterior é calculado através da equação (3.14) [8]. Na utilização desta equação varia o coeficiente global de transmissão de calor e a área, dependendo do tipo e das dimensões do elemento em análise. A temperatura interior geralmente é um valor ou intervalo de valores fixos, enquanto a temperatura exterior é um valor que pode ser fixo, numa análise de projecto, embora na análise anual o valor varia consoante a hora do dia.

$$q_{\text{env,ext}} = UA(T_{\text{out}} - T_{\text{in}}) \quad (3.14)$$

Em que:

$q_{\text{env,ext}}$  – Ganho de calor através da envolvente externa [W];

$U$  – Coeficiente global de transmissão de calor [W/(m<sup>2</sup>.K)];

$A$  – Área [m<sup>2</sup>];

$T_{\text{out}}$  – Temperatura exterior [°C];

$T_{\text{in}}$  – Temperatura interior [°C].

### 3.3.2. Ganho de calor através da envolvente interior

Nos ganhos de calor através da envolvente interior, as variações que ocorrem no cálculo utilizando a equação (3.15) são iguais ao já mencionado no ganho de calor através da envolvente exterior, só com a diferença que a temperatura exterior é substituída pela temperatura do espaço adjacente [8].

$$q_{env,int}=UA(T_b-T_{in}) \quad (3.15)$$

Em que:

$q_{env,int}$  – Ganho de calor através da envolvente interna [W];

$U$  – Coeficiente global de transmissão de calor [W/(m<sup>2</sup>.K)];

$A$  – Área [m<sup>2</sup>];

$T_b$  – Temperatura média do ar do espaço adjacente [°C];

$T_{in}$  – Temperatura interior [°C].

### 3.3.3. Ocupação

As cargas térmicas devido à ocupação, servem para quantificar o calor produzido pelo ser humano num determinado espaço. Estas cargas dependem do metabolismo do ser humano e do tipo de actividade a decorrer no espaço. As cargas de ocupação estão divididas em carga sensível e carga latente, uma vez que o ser humano além de produzir calor que aumenta a temperatura do espaço também produz calor que não aumenta a temperatura, mas que se traduz na produção de vapor de água no espaço. Caso o número de pessoas a ocupar o espaço seja conhecido, o cálculo do calor sensível devido à ocupação é realizado através da equação (3.16) [7].

$$q_{ocup,s}=q_{s,per}N \quad (3.16)$$

No cálculo do calor latente devido à ocupação, é utilizada a equação (3.17) [7].

$$q_{ocup,l}=q_{l,per}N \quad (3.17)$$

Caso não seja conhecido o número de pessoas, mas sim a densidade de ocupação a equação (3.18) traduz a mudança da densidade de ocupação em número de ocupantes.



$$N = \frac{1}{d_n} \times A \quad (3.18)$$

Em que:

$q_{\text{ocup,s}}$  – Carga sensível devida aos ocupantes [W];

$q_{\text{s,per}}$  – Carga sensível por ocupante [W/pessoa];

$N$  – Número de ocupantes;

$q_{\text{ocup,l}}$  – Carga latente devida aos ocupantes [W].

$q_{\text{l,per}}$  – Carga latente por ocupante [W/pessoa].

$d_n$  – Densidade de ocupação [ $\text{m}^2$ /pessoa];

$A$  – Área do espaço condicionado [ $\text{m}^2$ ].

Na Tabela 3.3 são apresentados os valores referentes ao calor sensível e latente consoante o local e a actividade de ocupação. De referir que os valores ajustados na coluna “Calor total” baseiam-se na percentagem normal de homens, mulheres e crianças para as actividades listadas.

Tabela 3.3. Calor sensível e latente de ocupantes

Actividade	Local	Calor total [W]		Calor sensível [W]	Calor latente [W]
		Masculino	Ajustado		
Sentado	Teatro, matiné	115	95	65	30
Sentado	Teatro, noite	115	105	70	35
Sentado, trabalho leve	Escritórios, hotéis, apartamentos	130	115	70	45
Trabalho moderado de escritório	Escritórios, hotéis, apartamentos	140	130	75	55
Em pé, trabalho leve, andar	Armazém, retalho	160	130	75	55
Andar, em pé	Drogaria, banco	160	145	75	70
Trabalho sedentário	Restaurante	145	160	80	80
Trabalho leve de bancada	Fábrica	235	220	80	140
Dança moderada	Salão de dança	265	250	90	160
Andar (4,8 km/h), trabalho com maquinaria leve	Fábrica	295	295	110	185
Jogar Bowling	Pista de Bowling	440	425	170	255
Trabalho pesado	Fábrica	440	425	170	255
Trabalho com maquinaria pesada	Fábrica	470	470	185	285
Desporto	Ginásio	585	525	210	315

### 3.3.4. Iluminação

A potência total de iluminação é obtida através da classificação das lâmpadas instaladas, sem a inclusão dos balastros. Para a obtenção da carga térmica devido à iluminação, a potência total de iluminação é multiplicada por dois factores. O primeiro factor é o factor de utilização, que é a relação entre a potência utilizada com a potência total instalada. O segundo factor é o de compensação, que relaciona o consumo de potência dos equipamentos de iluminação incluindo lâmpadas e balastro, com o consumo de potência nominal das lâmpadas [7]. Assim sendo, a carga térmica devido à iluminação é calculada de acordo com a equação (3.19).

$$q_{ilu} = W F_{ul} F_{sa} \quad (3.19)$$

Em que:

$q_{ilu}$  – Carga térmica devido à iluminação [W];

$W$  – Potência total de iluminação [W];

$F_{ul}$  – Factor de utilização;

$F_{sa}$  – Factor de compensação.

Caso a informação luminotécnica do espaço seja unicamente a sua densidade de iluminação, o cálculo da carga térmica devido à iluminação é determinado através da equação (3.20) [9].

$$q_{ilu} = W_{1A} A_N \quad (3.20)$$

Em que:

$W_{1A}$  – Densidade de iluminação [W/m<sup>2</sup>];

$A$  – Área do espaço condicionado [m<sup>2</sup>].

### 3.3.5. Carga devida a equipamentos com motor eléctrico

Segundo AHSRAE (2013), a carga térmica devida a equipamentos com motor eléctrico é calculada através da equação (3.21) [7]. Esta equação é aplicada se o motor e o equipamento estão no espaço em análise.

$$q_{eq.ele} = \frac{P}{E_M} F_{UM} F_{LM} \quad (3.21)$$

Em que:

$q_{eq.ele}$  – Carga devida a equipamentos com motor eléctrico [W];

$P$  – Potência do motor [W];

$E_M$  – Eficiência do motor;

$F_{UM}$  – Factor de utilização do motor;

$F_{LM}$  – Factor de carga do motor;

O factor de utilização do motor pode ser aplicado, quando é conhecido o seu horário de funcionamento. Nos casos de utilizações convencionais, o valor do coeficiente de utilização assume o valor de um. Nos outros casos (utilizações intermitentes) este factor tem o valor inferior a um. O factor de carga, relaciona a fracção de carga nominal que o equipamento eléctrico emite nas condições da estimativa da carga térmica. O valor do factor de carga normalmente é igual a um, podendo também ser inferior [7].

### 3.3.6. Carga devida a aplicações diversas

A carga térmica devida a aplicações diversas é a carga introduzida no espaço que não é enquadrada nem em carga térmica devida a iluminação, a ocupação, e de equipamentos eléctricos. Um exemplo prático de uma carga térmica diversa, são as cargas originadas no acto de cozinhar, onde existe tanto carga térmica sensível e latente. No entanto, não podem ser enquadradas nos tipos de cargas anteriormente descritas.

Segundo ASHRAE (2013), a carga térmica sensível é calculada através da equação (3.22) [7].

$$q_{div,s} = q_{input} F_L \quad (3.22)$$

Em que:

$q_{div,s}$  – Carga sensível da aplicação [W];

$q_{input}$  – Energia trocada pela aplicação [W];

$F_L$  – Factor de carga da aplicação.

O factor de carga da aplicação,  $F_L$ , é um factor ponderativo que define a porção sensível da carga total da aplicação, sendo que este factor depende da fonte de energia da aplicação

[7]. A Tabela 3.4 apresenta os valores típicos de factores de carga de algumas cargas diversas.

Tabela 3.4. Factores de carga de algumas cargas diversas [7].

Aplicação	Factor de carga	
	Eléctrico / Vapor	Gás
Grelhador	0,07	0,06
Fritadeira	0,03	0,02
Forno por convecção	0,07	0,08
Churrasqueira	0,24	0,11
Cozedura a vapor	0,04	-

### 3.3.7. Carga devida à infiltração de ar

As cargas térmicas devido às infiltrações de ar, teoricamente é apenas necessário conhecer o caudal volúmico infiltrado no espaço condicionado e as condições interiores e exteriores ao espaço [5]. A incerteza no cálculo desta tipologia de carga térmica reside no facto de não ser possível conhecer as condições de utilização de janelas e portas do espaço [5]. Este tipo de carga térmica pode ser dividida em sensível e latente. A componente sensível da carga térmica devido a infiltrações de ar pode ser calculada de acordo com a equação (3.23) [8]. Para a determinação da componente latente é utilizada a equação (3.24) [8]. Admitindo-se que a massa específica do ar é constante, a carga total devido à infiltração de ar é a soma da carga sensível e latente. Podendo também ser feito o cálculo da carga total através da equação (3.25) [8].

$$q_{ar,s}=1,23\dot{V}(T_{out}-T_{in}) \quad (3.23)$$

$$q_{ar,l}=3010\dot{V}(\omega_{out}-\omega_{in}) \quad (3.24)$$

$$q_{ar,total}=1,20\dot{V}(h_{out}-h_{in}) \quad (3.25)$$

Em que:

$q_{ar,s}$  – Carga sensível devido à ventilação e infiltração de ar [W];

$\dot{V}$  – Caudal volúmico de ar [m<sup>3</sup>/s];

$T_{out}$  – Temperatura exterior [°C];

$T_{in}$  – Temperatura interior [°C];

$q_{ar,l}$  – Carga latente devido à ventilação e infiltração de ar [W];  
 $\omega_{out}$  – Humidade específica do ar exterior [kg<sub>v</sub>/kg<sub>a</sub>];  
 $\omega_{in}$  – Humidade específica do ar interior [kg<sub>v</sub>/kg<sub>a</sub>];  
 $q_{ar, total}$  – Carga total devido à ventilação e infiltração de ar [W];  
 $h_{out}$  – Entalpia do ar exterior [kJ/kg];  
 $h_{in}$  – Entalpia do ar interior [kJ/kg].

### 3.3.8. Carga térmica de arrefecimento e aquecimento

Estando definidas as parcelas constituintes de uma carga térmica de arrefecimento e aquecimento, deve-se quantificar a carga térmica na sua globalidade não apenas nas suas parcelas. Assim sendo, a carga térmica de arrefecimento traduz-se no somatório de todas as parcelas anteriormente descritas. As equações (3.26), (3.27) e (3.28) representam a carga térmica de arrefecimento sensível, latente e total respectivamente.

$$q_{total,s} = q_{env,ext} + q_{env,int} + q_{ocup,s} + q_{ilu} + q_{eq,ele} + q_{div,s} + q_{ar,s} \quad (3.26)$$

$$q_{total,l} = q_{ocup,l} + q_{div,l} + q_{ar,l} \quad (3.27)$$

$$q_{total} = q_{total,s} + q_{total,l} \quad (3.28)$$

No cálculo da carga térmica de aquecimento, unicamente são contabilizadas as perdas térmicas do espaço. Sendo o seu valor igual ao somatório das perdas de calor causadas por transmissão térmica e infiltrações de ar. As equações (3.29), (3.30) e (3.31) representam a carga térmica de aquecimento sensível, latente e total respectivamente

$$q_{total,s} = q_{env,ext} + q_{env,int} + q_{ar,s} \quad (3.29)$$

$$q_{total,l} = q_{ar,l} \quad (3.30)$$

$$q_{total} = q_{total,s} + q_{total,l} \quad (3.31)$$

Em que:

$q_{total,s}$  – Carga térmica de arrefecimento / aquecimento sensível [W];  
 $q_{total,l}$  – Carga térmica de arrefecimento / aquecimento latente [W];

$q_{total}$  – Carga térmica de arrefecimento / aquecimento [W];  
 $q_{cpc}$  – Carga térmica devido à envolvente exterior [W];  
 $q_{dtc}$  – Carga térmica devido à envolvente interior [W];  
 $q_{ocup,s}$  – Carga térmica sensível devido à ocupação [W];  
 $q_{ocup,l}$  – Carga térmica latente devido à ocupação [W];  
 $q_{ilu}$  – Carga térmica devido à iluminação [W];  
 $q_{eq,ele}$  – Carga térmica devido a equipamentos eléctricos [W];  
 $q_{div,s}$  – Carga térmica sensível devido a aplicações diversas [W];  
 $q_{div,l}$  – Carga térmica latente devido a aplicações diversas [W];  
 $q_{ar,s}$  – Carga térmica sensível devido à infiltração de ar [W];  
 $q_{ar,l}$  – Carga térmica latente devido à infiltração de ar [W].

### 3.4. Psicrometria

Segundo Monteiro (2015), a psicrometria estuda as misturas de ar e vapor de água [2]. Aprofundando um pouco mais, a Psicrometria dedica-se à análise das propriedades físicas e termodinâmicas das misturas entre gases e vapor e suas aplicações práticas, como, por exemplo [2]:

- O controlo climático, em instalações de condicionamento de ar para conforto térmico;
- Evitar a condensação em superfícies frias;
- O arrefecimento evaporativo.

#### 3.4.1. Humidade relativa

A humidade relativa,  $\varphi$ , é a relação entre a pressão parcial real do vapor de água contida no ar húmido e a pressão do vapor saturado à mesma temperatura [2].

$$\varphi = \frac{p_v}{p_s} \times 100 \quad (3.32)$$

Em que:

$p_v$  – Pressão parcial do vapor [Pa];  
 $p_s$  – Pressão de saturação [Pa].

A humidade relativa é, a par da temperatura, um parâmetro importante para a construção do conforto térmico e deve-se situar entre os 40 e 60% [2].

### 3.4.2. Humidade específica

A humidade específica,  $\omega$ , representa o indicador da composição do ar húmido de uma determinada quantidade de ar [2]. Isto é, que o conteúdo em humidade é definido pela razão entre a massa de vapor de água, e massa de ar seco, presente na mistura [2]. A equação (3.33) estabelece a relação entre a humidade específica e a pressão de vapor [23].

$$\omega = 0,622 \frac{p_v}{p - p_v} \quad (3.33)$$

Em que:

$p$  – Pressão atmosférica, que tem um valor de 101325 [Pa];

$p_v$  – Pressão de vapor [Pa].

Para contemplar o efeito da altitude na pressão são utilizadas duas equações, sendo elas a equação (3.34) e a (3.35) [24]. A equação (3.34) é utilizada para determinar a altitude a uma determinada pressão, enquanto a equação (3.35) determina a pressão atmosférica a uma determinada pressão [24].

$$H = H_b - \frac{T_{bm}}{L_b} \left[ \left( \frac{P}{P_b} \right)^{\frac{-R \cdot L_b}{g_0 \cdot M}} - 1 \right] \quad (3.34)$$

$$p = P_b \cdot \left[ 1 + \frac{L_b}{T_{bm}} \cdot (H - H_0) \right]^{\frac{-g_0 \cdot M}{R \cdot L_b}} \quad (3.35)$$

Em que:

$P_b$  – Pressão estática (ao nível do mar) [Pa];

$T_{bm}$  – Temperatura normalizada (ao nível do mar) [K];

$L_b$  – Taxa de lapso da temperatura, que tem um valor de 0,0065 [K/m];

$H$  – Altura sobre o nível do mar [m];

$H_b$  – Altura na camada inferior da atmosfera [m];

$R$  – Constante universal de um gás, que tem um valor de 8,31432 [(N.m)/(mol.K)];

$g_0$  – Constante de aceleração gravitacional, que tem um valor de 9,80665 [m/s<sup>2</sup>];

M – Massa molar do ar, que tem um valor de 0,0289644 [kg/mol].

### 3.4.3. Entalpia

A entalpia é a propriedade termodinâmica que mede a energia térmica (latente mais sensível) de uma determinada massa de ar, acima de determinada temperatura de referência (0 °C), representando a energia de 1 kg de ar seco [2]. A entalpia do ar seco é determinada utilizando a equação (3.36) [23].

$$h_a = 1,006(T - 0) \quad (3.36)$$

A equação que determina a entalpia de vapor é a equação (3.37) [23].

$$h_v = 2500,7 + 1,84T \quad (3.37)$$

A entalpia do ar é calculada de acordo com a equação (3.38) [23].

$$h = h_a + \omega h_v \quad (3.38)$$

Em que:

$h_a$  – Entalpia do ar seco [kJ/kg];

$h_v$  – Entalpia do vapor de água [kJ/kg];

$h$  – Entalpia do ar [kJ/kg];

$T$  – Temperatura [°C];

$\omega$  – Umidade específica [kg<sub>v</sub>/kg<sub>a</sub>].



## 4. Programação desenvolvida

Neste capítulo é abordado o trabalho realizado com vista ao desenvolvimento da aplicação informática para o cálculo da condução térmica em edifícios.

A aplicação informática foi realizada utilizando o software de cálculo computacional MATrix LABoratory (MATLAB). A aplicação é constituída por três etapas, sendo elas a estrutura de dados, funções e ambiente gráfico. Irão ser abordados os vários tipos de estruturas de dados constituintes da aplicação como também serão abordadas as várias interfaces gráficas de utilizador (GUI) criadas. Durante a abordagem das interfaces gráficas são introduzidas as funções que são utilizadas na respectiva interface.

### 4.1. Estruturas de dados

A estrutura de dados da aplicação informática é composta pelos seguintes elementos: materiais de construção de edifícios, locais geográficos, vidros simples, duplos e triplos, caixilhos, janelas simples e duplas, portas, elementos construtivos e horários de ocupação, iluminação e de equipamentos eléctricos. É de referir que as estruturas de dados apresentadas neste subcapítulo contêm informação pré-definida na aplicação, tirando a possibilidade de editar ou eliminar esta informação. Os dados são armazenados em variáveis heterogénea de classe *struct*. Cada variável é uma lista, cuja dimensão corresponde ao número de elementos armazenados em cada uma. Na Tabela 4.1 é apresentada a informação definida na estrutura de dados dos elementos: materiais de construção de edifícios, locais geográficos, vidros simples, duplos e triplos, caixilhos, janelas simples e duplas, portas, elementos construtivos e horários de ocupação, iluminação e de equipamentos eléctricos.

Tabela 4.1. Descrição da estrutura de dados

<b>Estrutura de dados (descrição das variáveis)</b>	<b>Dados (descrição dos campos por variável)</b>
Locais geográficos	Nome
	Latitude
	Longitude
	Altitude
Materiais de construção de edifícios	Nome
	Classificação
	Massa volúmica
	Condutibilidade térmica
Vidros	Nome
	Tipo
	Coeficiente global de transmissão de calor
Caixilhos	Nome
	Coeficiente global de transmissão de calor
Janelas	Nome
	Tipo
	Coeficiente global de transmissão de calor
Portas	Nome
	Coeficiente global de transmissão de calor
Elementos construtivos: Paredes de fachada, pavimentos e coberturas	Nome
	Tipo
	Coeficiente global de transmissão de calor
	Resistência térmica interior
	Resistência térmica exterior
	Tabela com informação sobre a composição do elemento construtivo
Horários de ocupação, iluminação e de equipamentos eléctricos	Nome
	Tipo
	Tabela com os coeficientes do padrão de referência do espaço

#### 4.1.1. Materiais de construção de edifícios

Esta estrutura de dados armazena os materiais utilizados para construção de edifícios (Anexo A). A aplicação tem previamente armazenado dados relativos a 139 materiais distintos, agrupados em 17 classes: água, argamassas, betões, borrachas, fibrocimento, gases, gessos (estuques), materiais de impermeabilização e mástiques, inertes, solos e

terras, isolantes térmicos, madeiras e derivados, materiais plásticos, materiais cerâmicos, metais, pedras (naturais e artificiais), revestimentos de pisos ou paredes e vidros. A informação que cada elemento contém é o nome, classificação do material, massa volúmica e condutibilidade térmica.

Os valores de massa volúmica e condutibilidade térmica foram retirados da Informação Técnica de edifícios nº50 (ITE 50) do Laboratório Nacional de Engenharia Civil (LNEC) [3]. A Figura 4.1 representa o primeiro elemento da classe de betão na estrutura de dados de materiais de construção.

```
%Betão de inertes correntes (calcários, siliciosos e silico-calcários)

%Betão normal

material.betao(1).nome = 'Betão normal';
material.betao(1).classificacao = 'Betão de inertes correntes (calcários, siliciosos e silico-calcários';
material.betao(1).rho = 2150;           %kg/m^3 retirado:ITE 50
material.betao(1).k = 1.65;             %W/(m*°C) retirado:ITE 50
```

Figura 4.1. Primeiro elemento da classe de betão na estrutura de dados de materiais de construção.

#### 4.1.2. Locais

Esta estrutura de dados permite armazenar informação relativa à localização geográfica e é composta pelos campos: nome, latitude, longitude e altitude do local. As cidades definidas foram: Aveiro, Beja, Braga, Bragança, Castelo Branco, Coimbra, Évora, Faro, Funchal, Guarda, Leiria, Lisboa, Ponta Delgada, Portalegre, Porto, Santarém, Setúbal, Viana do Castelo, Vila Real, Viseu.

Os dados de latitude, longitude e altitude dos diversos locais foram obtidos com recurso à aplicação CLIMA-SCE - Software para o Sistema Nacional de Certificação de Edifícios do LNEC. A Figura 4.2 representa o primeiro elemento na estrutura de dados de locais.

```
% Aveiro
clima(1).nome='Aveiro';
clima(1).latitude='40,6°N (nominal)';
clima(1).longitude='8,6°W (nominal)';
clima(1).altitude= '50';
```

Figura 4.2. Primeiro elemento na estrutura de dados de locais.

#### 4.1.3. Vidros simples, duplos e triplos

A estrutura de dados de vidros é composta por 5 elementos. Os valores são retirados da tabela 4 do capítulo 15 do ASHRAE Handbook – Fundamentals 2013 [7]. Os valores retirados foram para o vidro simples de 3 mm de espessura, vidro duplo e triplo de 6 mm de espaço de ar, e vidro duplo e triplo com 13 mm de espaço de ar. Os valores de espessura dos vidros são considerados como sendo igual a 3 mm cada.

A informação que cada elemento contém é o nome, tipo de vidro e o coeficiente global de transmissão de calor do vidro. A Figura 4.3 representa o primeiro elemento da estrutura de dados de vidros.

```
vidro(1).nome = 'Vidro simples:3 mm Vidro';  
vidro(1).tipo = 'Vidro simples';  
vidro(1).ug = 5.91; %W/(m^2.K)
```

Figura 4.3. Primeiro elemento da estrutura de dados de vidros.

#### 4.1.4. Caixilhos

A estrutura de dados de caixilhos é composta por 5 elementos. Os valores do coeficiente global de transmissão de calor de cada elemento foram retirados do Anexo D da norma BS EN 10077-1:2006 [21]. Os caixilhos considerados foram os caixilhos de madeira, poliuretano e de PVC.

A informação que cada elemento contém é o nome e o coeficiente global de transmissão de calor do caixilho. A Figura 4.4 representa o primeiro elemento da estrutura de dados de caixilhos.

```
% Base de dados de caixilhos  
  
caixilho(1).nome = 'Caixilho de madeira(densidade = 700 kg/m^3 ,cond.térmica = 0.18 W/(mk),espessura = 50 mm) ';  
caixilho(1).uf = '2'; %W/(m^2K)
```

Figura 4.4. Primeiro elemento da estrutura de dados de caixilhos.

#### 4.1.5. Janelas

A estrutura de dados de janelas simples e duplas é composta por 21 elementos. Os valores dos coeficientes globais de transmissão de calor da janela foram retirados dos Quadros III.1, III.2 e III.3 do Anexo III da Informação Técnica de Edifícios nº 50 (ITE 50) do Laboratório Nacional de Engenharia Civil (LNEC) [3].

A informação que cada elemento contém é o nome, tipo e o coeficiente global de transmissão de calor da janela. A Figura 4.5 representa o primeiro elemento da estrutura de dados de janelas.

```
% Caixilharia em Madeira
janela(1).nome='Vão envidraçado vertical com caixilharia em madeira(Janela fixa,
giratória ou de correr com vidro simples)';
janela(1).tipo='Janela simples';
janela(1).uw='5.1'; %W/(m^2.K)
```

Figura 4.5. Primeiro elemento da estrutura de dados de janelas.

#### 4.1.6. Portas

A estrutura de dados relativa a portas é composta por 5 elementos. Os valores dos coeficientes globais de transmissão de calor foram retirados da tabela 6 e 7 do capítulo 15 do ASHRAE Handbook – Fundamentals 2013 [7].

A informação que cada elemento contém é o nome e o coeficiente global de transmissão de calor da porta. A Figura 4.6 representa o primeiro elemento da estrutura de dados de portas.

```
porta(1).nome='Porta giratória - 3 folhas (2.44 x 2.13 m)';
porta(1).ud=4.46;
```

Figura 4.6. Primeiro elemento da estrutura de dados de portas.

#### 4.1.7. Elementos construtivos: paredes de fachada, pavimentos e coberturas

A estrutura de dados de elementos construtivos é composta por 23 elementos. Os valores dos coeficientes globais de transmissão de calor foram retirados do Anexo II da Informação Técnica de Edifícios nº 50 (ITE 50) do Laboratório Nacional de Engenharia Civil (LNEC) [3].

A informação que cada elemento contém é o nome, tipo e o coeficiente global de transmissão de calor da janela, resistência térmica interior, resistência térmica exterior e informação sobre a sua composição. A Figura 4.5 representa o primeiro elemento da estrutura de dados de elementos construtivos.

```
% Paredes de Fachada simples sem isolamento térmico
elementos(1).nome = 'Paredes simples de fachada sem isolamento térmico: tijolo
furado';
elementos(1).tipo = 'Parede';
elementos(1).u = 1.3;
elementos(1).rsi = '-';
elementos(1).rse = '-';
elementos(1).tabela = {'-', 'Tijolo furado', '-', '-', '200 a 240'};
```

Figura 4.7. Primeiro elemento da estrutura de dados de elementos construtivos.

#### 4.1.8. Horários de ocupação, iluminação e de equipamentos eléctricos

Os horários de ocupação, iluminação e de equipamentos eléctricos foram retirados do Anexo XV do Decreto-Lei n. °79/2006 de 4 de Abril [25]. A legislação utilizada na obtenção dos padrões de referência é apenas de referência não sendo uma legislação obrigatória. Os padrões de referência retirados foram sobre as seguintes tipologias: Hipermercado, Venda por grosso, Centro comercial, Pequenas lojas, Restaurantes, Pastelarias, Cinemas e teatros, Escritórios, Bibliotecas, Museus e galerias, Tribunais, Ministérios e Câmaras, Estabelecimentos de ensino, Estabelecimento de ensino superior, Estabelecimento de saúde sem internamento, Estabelecimento de saúde com internamento.

A informação que cada elemento contém é o nome e data. O campo data contém o horário da respectiva tipologia de edifício, que é composto por 24 horas (correspondendo às linhas do campo “data”) em 7 dias da semana mais o dia feriado (correspondendo às colunas do campo “data”). A Figura 4.8 representa os elementos na estrutura de dados de ocupação, iluminação e equipamentos eléctricos.



Fields	 nome	 data
1	'Hipermercado'	24x8 double
2	'Venda por grosso'	24x8 double
3	'Centro comercial'	24x8 double
4	'Pequenas lojas'	24x8 double
5	'Restaurantes'	24x8 double
6	'Pastelarias'	24x8 double
7	'Cinemas e teatros'	24x8 double
8	'Escritórios'	24x8 double
9	'Bibliotecas'	24x8 double
10	'Museus e galerias'	24x8 double
11	'Tribunais, Ministérios e Câmaras'	23x8 double
12	'Estabelecimentos de ensino'	24x8 double
13	'Estabelecimentos de ensino superior'	24x8 double
14	'Estabelecimentos de saúde sem internamento'	24x8 double
15	'Estabelecimento de saúde com internamento'	24x8 double

Figura 4.8. Elementos na estrutura de dados de ocupação, iluminação e equipamentos eléctricos.

## 4.2. Ambiente gráfico

A aplicação para o cálculo da condução térmica em edifícios é composta por 18 interfaces gráficas. Para estabelecer a ligação entre as interfaces gráficas foi criada uma barra de menus (com três níveis) para navegar entre cada uma delas. A Tabela 4.2 especifica os diversos níveis de menus da aplicação. O primeiro nível é enumerado através da numeração árabe. O segundo nível de menus é enumerado pelo alfabeto. O terceiro nível é enumerado pela numeração romana. Só foram criadas as interfaces gráficas finais (menus que estão a azul na Tabela 4.2), os restantes menus foram criados apenas para uma melhor hierarquização na navegação do utilizador.

Tabela 4.2. Níveis de menus da aplicação.

Primeiro nível de menus	Segundo nível de menus	Terceiro nível de menus
1. Início		
2. Local		
3. Material		
4. Elementos construtivos	a. Vidro b. Caixilho c. Janela	i. Adicionar janela simples ii. Adicionar janela dupla iii. Adicionar janela de dupla folha no mesmo aro iv. Ver/eliminar janela
	d. Porta	i. Adicionar porta ii. Ver/eliminar porta
	e. Paredes, tectos, coberturas e pavimentos	i. Adicionar ii. Ver/eliminar
5. Espaço	a. Definir espaço b. Ver/eliminar	
6. Horários	a. Adicionar horário b. Definir horário	
7. Relatórios		

#### 4.2.1. Gui principal

O GUI principal (Figura 4.9) permite criar, carregar, importar, exportar e eliminar projectos. Na função de abertura são lidas e mostradas as imagens dos logótipos do Instituto Superior de Engenharia de Lisboa (ISEL) e da Área Departamental de Engenharia Mecânica (ADEM), conforme se pode visualizar na Figura 4.9.



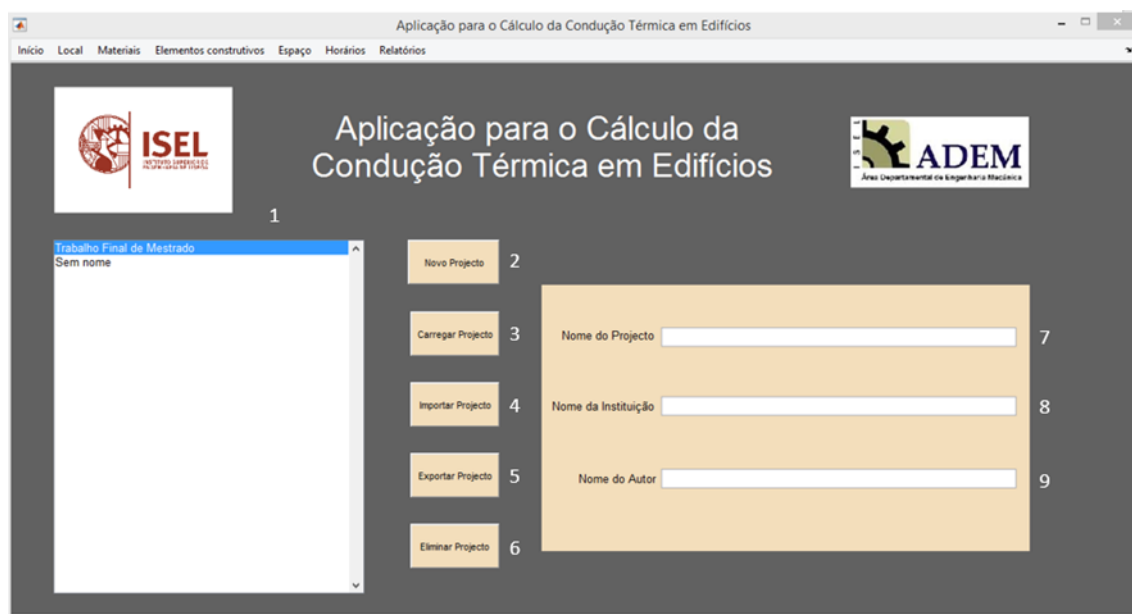


Figura 4.9. GUI principal.

Legenda:

- |                                     |                                            |
|-------------------------------------|--------------------------------------------|
| 1 – Caixa de listagem: “Projectos”; | 6 – Botão: “Eliminar Projecto”;            |
| 2 – Botão: “Novo Projecto”;         | 7 – Caixa de texto: “Nome do Projecto”;    |
| 3 – Botão: “Carregar Projecto”;     | 8 – Caixa de texto: “Nome da Instituição”; |
| 4 – Botão: “Importar Projecto”;     | 9 – Caixa de texto: “Nome do Autor”;       |
| 5 – Botão: “Exportar Projecto”;     |                                            |

Foi programado para que, quando esta janela for executada, seja carregada para o espaço de variáveis um vector com o nome “projecto”, que contem a lista dos projectos criados pelo utilizador na aplicação, e que se encontra no ficheiro “projecto.mat”. Essa listagem é disponibilizada para o utilizador numa caixa de listagem (1). No caso de não existirem projectos é enviada para a lista de projectos a mensagem, “Não existem projectos”.

Caso seja pressionado o botão “Novo Projecto” (2) é adicionada mais um elemento à variável “projecto”, sendo carregadas as diversas estruturas de dados descritas no subcapítulo 4.1. O nome do projecto é definido automaticamente a partir das informações introduzidas nas caixas de texto “Nome do Projecto” (7), “Nome da Instituição” (8) e “Nome do Autor” (9). O texto introduzido nestas caixas é relacionado de modo lógico e armazenado no novo elemento na nova linha da variável ‘projecto’. Caso estas caixas de texto não estejam preenchidas a aplicação define como “Nome do Projecto”, “Nome da Instituição” e “Nome do Autor” como “Sem nome”, “Sem instituição” e sem “Sem autor”,

respectivamente. Para finalizar a rotina deste botão a lista de projectos armazenados na variável “projecto” é utilizada para actualizar a informação apresentada na caixa de listagem (1).

A acção de pressionar o botão “Carregar Projecto” (3) cria uma variável global com o nome “carproj”. Esta variável define o projecto que foi carregado em toda a aplicação informática. Consoante o valor da variável “carproj” é enviado para os campos 7, 8 e 9 o nome do projecto, da instituição e do autor desse projecto.

O botão “Exportar Projecto” (5) permite que, caso existam projectos na lista de projectos, o seleccionado seja exportado para um ficheiro de dados do MATLAB (extensão “.mat”), caso não existam projectos é criada a mensagem de erro a dizer “Não existem projectos”. O botão “Importar Projecto” (4), adiciona uma linha à variável “projecto” com a informação de um projecto anteriormente exportado. Tanto na importação como na exportação de projectos para finalizar a rotina é actualizada a caixa de listagem “Projectos” (1) no GUI principal.

O botão “Eliminar Projecto” (6) elimina o projecto que foi seleccionado na lista de projectos. De seguida é enviada a nova lista com os nomes dos projectos contidos na variável ‘Projecto’ para a lista de projectos. Caso não exista nenhuma entrada, é enviada a mensagem “Não existem projectos” para a caixa de listagem “Projectos” (1).

#### **4.2.2. GUI de definição do local**

O GUI de definição do local (Figura 4.10) permite adicionar e eliminar locais à estrutura de dados. Os locais que podem ser eliminados são unicamente os introduzidos pelo utilizador. Este ambiente gráfico permite também definir o local da simulação numérica e carregar o ficheiro climático. A Figura 4.10 representa a interface gráfica que permite definir o local da aplicação informática para o cálculo da condução térmica em edifícios.

Quando a interface gráfica do local é iniciada, é enviado para o menu *popup* “Local” (1) a lista com os nomes dos locais que estão na estrutura de dados de projecto.

Para adicionar um novo local é criada uma linha à estrutura de dados de locais da variável “projecto”, onde são guardados os dados referentes as caixas de textos “Local” (Figura 14), “Latitude” (15), “Longitude” (16) e “Altitude” (17). Através do menu *popup* “Local” (1) é apresentado automaticamente nas caixas de texto 2, 3 e 4 a informação sobre a

“Latitude”, “Longitude” e “Altitude”, respectivamente ao local seleccionado armazenado na estrutura de dados, após selecção da localização.

Figura 4.10. GUI de definição do local

Legenda:

- |                                                                          |                                                 |
|--------------------------------------------------------------------------|-------------------------------------------------|
| 1 – Menu <i>popup</i> : “Local”;                                         | 10 – Caixa de texto: “Ficheiro Climático”;      |
| 2 – Caixa de texto: “Latitude”;                                          | 11 – Botão: “Carregar ficheiro climático”;      |
| 3 – Caixa de texto: “Longitude”;                                         | 12 – Botão: “Definir local”;                    |
| 4 – Caixa de texto: “Altitude”;                                          | 13 – Botão: “Eliminar local”;                   |
| 5 – Caixa de texto: “Temperatura de bolbo seco de projecto (Verão)”;     | 14 – Caixa de texto: “Novo Local”;              |
| 6 – Caixa de texto: “Temperatura de bolbo húmido de projecto (Verão)”;   | 15 – Caixa de texto: “Latitude do novo local”;  |
| 7 – Caixa de texto: “Variação diária de temperatura”;                    | 16 – Caixa de texto: “Longitude do novo local”; |
| 8 – Caixa de texto: “Temperatura de bolbo seco de projecto (Inverno)”;   | 17 – Caixa de texto: “Altitude do novo local”;  |
| 9 – Caixa de texto: “Temperatura de bolbo húmido de projecto (Inverno)”; | 18 – Botão: “Guardar novo local”                |

Quando seleccionado o botão “Carregar Ficheiro Climático” (11) é possível carregar o ficheiro climático do local. Este ficheiro pode ser importado a partir de dados em formato “dat” ou “epw”. Para realizar esta importação foram desenvolvidas duas funções “import\_epw” e “import\_dat” que retiram a informação necessária dos ficheiros climáticos e a guardam na variável “projecto”. Os ficheiros climáticos dos locais pré-definidos também devem de ser carregados. Esses ficheiros estão disponíveis quando é accionado o botão “Carregar ficheiro climático” (11).

Para definir o local são guardados os campos de: “Local”, “Latitude”, “Longitude”, “Altitude”, “Temperatura de bolbo seco de projecto (Verão)”, “Temperatura de bolbo húmido de projecto (Verão)”, “Variação diária de temperatura”, “Temperatura de bolbo seco de projecto (Inverno)”, “Temperatura de bolbo húmido de projecto (Inverno)”.

### 4.2.3. GUI de Materiais

O GUI de materiais (Figura 4.11) permite adicionar, ver a informação e eliminar um determinado material. De referir que só é possível eliminar os materiais introduzidos pelo utilizador. A Figura 4.11 representa a interface gráfica de materiais da aplicação informática para o cálculo da condução térmica em edifícios.

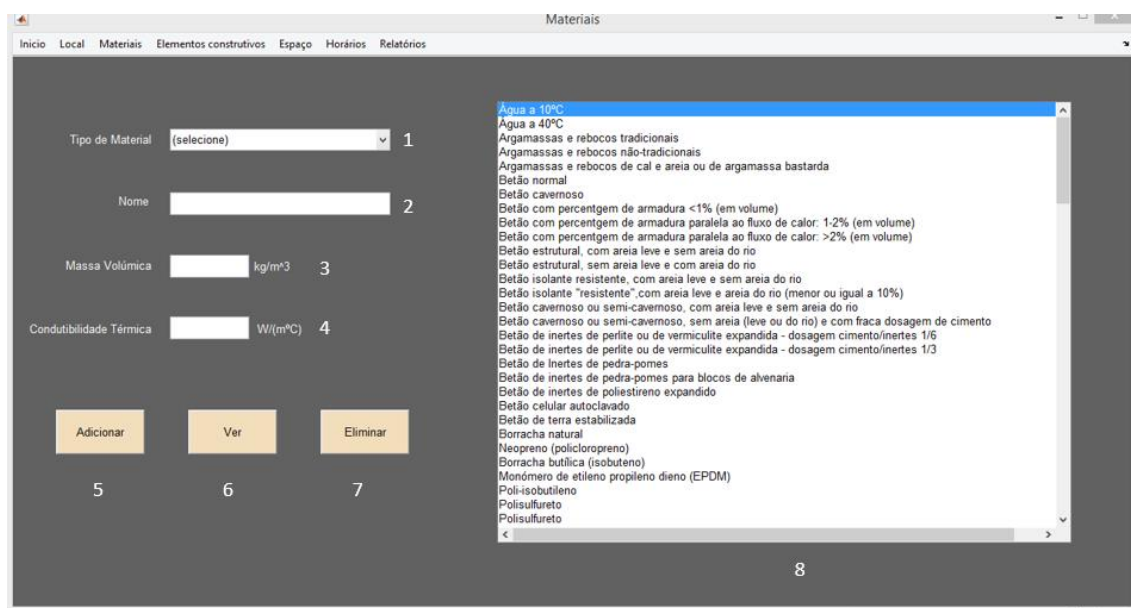


Figura 4.11. GUI de adicionar, ver e eliminar materiais.

Legenda:

- |                                                            |                                                                |
|------------------------------------------------------------|----------------------------------------------------------------|
| 1 – Menu <i>popup</i> : “Tipo de Material”;                | 5 – Botão: “Adicionar material”;                               |
| 2 – Caixa de texto: “Nome do material”;                    | 6 – Botão: “Ver material”;                                     |
| 3 – Caixa de texto: “Massa Volúmica do material”;          | 7 – Botão: “Eliminar material”;                                |
| 4 – Caixa de texto: “Condutibilidade Térmica do material”; | 8 – Caixa de listagem: “Materiais de construção de edifícios”. |

Na função de abertura é enviado para a caixa de listagem: “Materiais de construção de edifícios” (8) o nome de todos os materiais de todas as classes. Sempre que seleccionado o tipo de material no menu *popup* ‘Tipo de Material’ (1) é aplicado um filtro na lista de materiais existente na variável de estrutura de dados. Este filtro permite que apenas apareça na caixa de listagem “Materiais de construção de edifícios” (8) o nome dos materiais dessa mesma classe.

Para adicionar um material, primeiro é preciso ser seleccionada uma classe no ‘Tipo de Material’ (1). Para a dada classe de material é adicionada uma linha na estrutura de dados do projecto com o “Nome”, “Classificação”, “Massa Volúmica” e “Condutibilidade Térmica”. Para finalizar a rotina do botão “Adicionar material”, é actualizada a caixa de listagem “Materiais de construção de edifícios” (8).

Pressionando o botão “Ver material” (6), consoante o tipo e o material seleccionado na caixa de listagem “Materiais de construção de edifícios” (8) são enviados os respectivos dados para as seguintes caixas de texto: “Nome” (2), “Massa Volúmica” (3) e “Condutibilidade Térmica” (4).

Caso seja accionado o botão “Eliminar material” (7), só é permitido eliminar materiais criados pelo utilizador. Para eliminar qualquer material, primeiro é necessário seleccionar o tipo de material que deseja eliminar. Para finalizar a rotina é actualizada a caixa de listagem: “Materiais de construção de edifícios” (8).

#### **4.2.4. GUI de vidros**

O GUI de vidros (Figura 4.12 e 4.13) permite adicionar, visualizar ou eliminar dados sobre: vidros simples, duplos ou triplos. Sendo dada a possibilidade de calcular ou

introduzir o valor do coeficiente global de transmissão de calor. O botão “Adicionar vidro” (Figura 4.12-2) disponibiliza os objectos gráficos, de acordo com a Figura 4.12, permitindo adicionar o vidro. Enquanto o botão “Ver / Eliminar Vidro” (Figura 4.12-3) disponibiliza o ambiente gráfico de acordo com a Figura 4.13.

Quando GUI de vidros (Figura 4.12 e 4.13) é executado, é eliminada toda a informação apresentada na tabela “Camadas do vidro” (Figura 4.12-11), sendo apresentada a lista de nomes da classe do material vidros no menu *popup* (Figura 4.12-8), correspondente à estrutura de dados do projecto carregado no GUI principal (Figura 4.9).

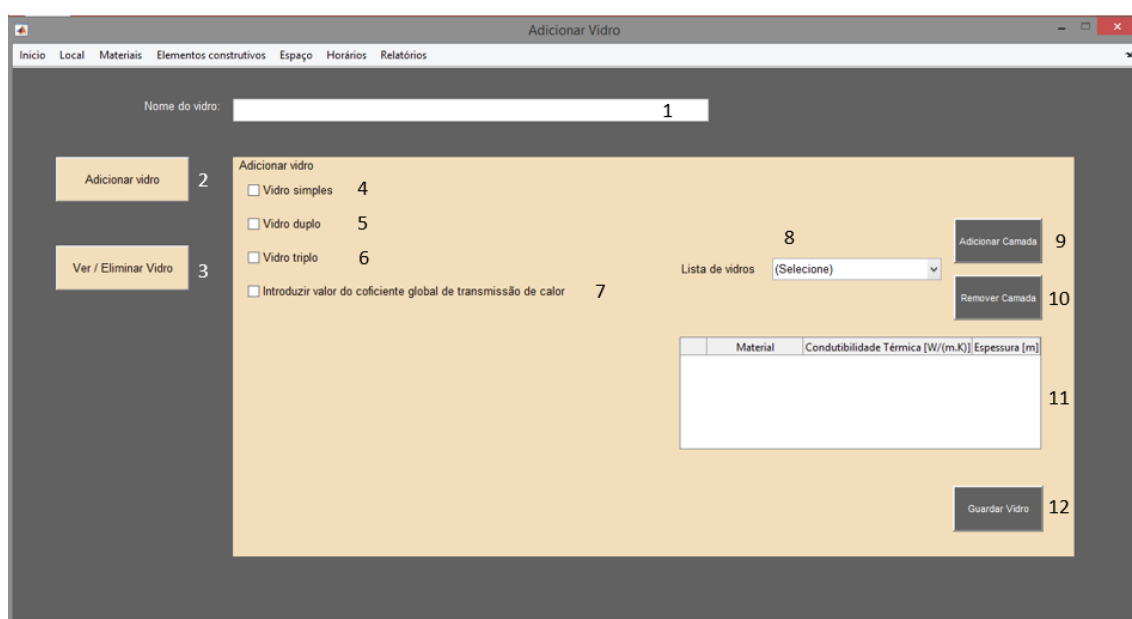


Figura 4.12. GUI de adicionar vidros.

#### Legenda:

- |                                            |                                                                                             |
|--------------------------------------------|---------------------------------------------------------------------------------------------|
| 1 – Caixa de texto: “Nome do vidro”;       | 7 – Caixa de verificação: “Introduzir valor do coeficiente global de transmissão de calor”; |
| 2 – Botão: “Adicionar vidro”;              | 8 – Menu <i>popup</i> : “Lista de vidros”;                                                  |
| 3 – Botão: “Ver/Eliminar Vidro”;           | 9 – Botão: “Adicionar Camada”;                                                              |
| 4 – Caixa de verificação: “Vidro simples”; | 10 – Botão: “Remover Camada”;                                                               |
| 5 – Caixa de verificação: “Vidro duplo”;   | 11 – Tabela: “Camadas do Vidro”;                                                            |
| 6 – Caixa de verificação: “Vidro triplo”;  | 12 – Botão: “Guardar Vidro”.                                                                |

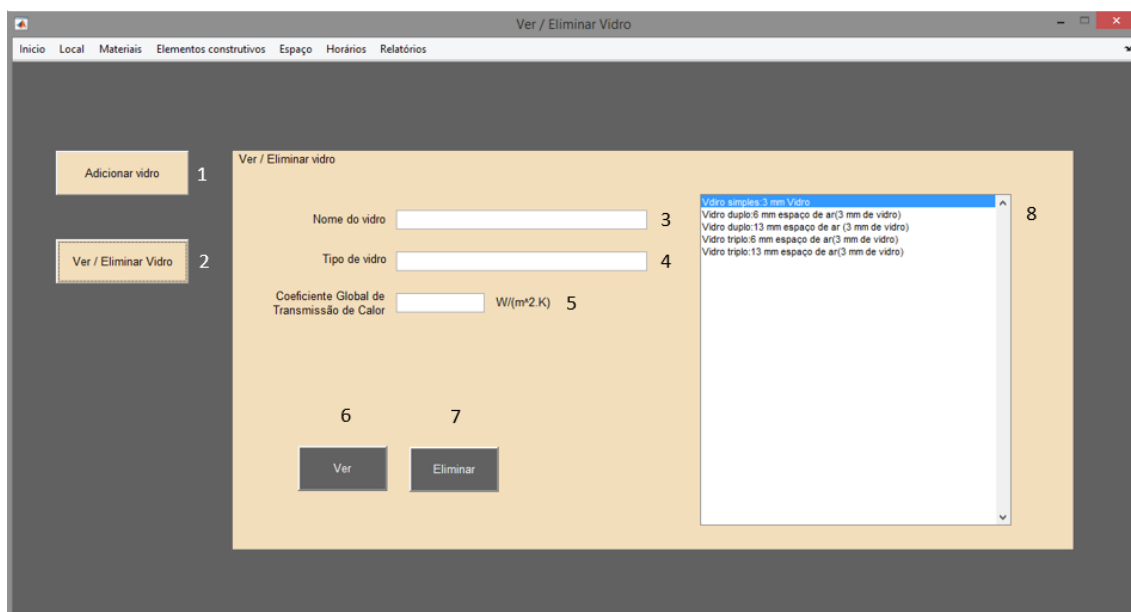


Figura 4.13. GUI de ver / eliminar vidros.

Legenda:

- |                                      |                                                                   |
|--------------------------------------|-------------------------------------------------------------------|
| 1 – Botão: “Adicionar vidro”;        | 5 – Caixa de texto: “Coeficiente Global de Transmissão de Calor”; |
| 2 – Botão: “Ver/Eliminar Vidro”;     | 6 – Botão: “Ver vidro”;                                           |
| 3 – Caixa de texto: “Nome do Vidro”; | 7 – Botão: “Eliminar vidro”;                                      |
| 4 – Caixa de texto: “Tipo de Vidro”; | 8 – Caixa de listagem: “Lista de vidros”.                         |

O botão “Adicionar camada” (Figura 4.12-9), adiciona uma linha à tabela “Camadas do vidro” (Figura 4.12-11) caso seja seleccionado o material no menu *popup* “Lista de vidros” (Figura 4.12-8). Essa linha tem o nome do material, a condutibilidade térmica e a espessura da camada do vidro. Caso não tenha sido seleccionado o material, aparece uma mensagem de erro a dizer “Escolha o material na lista de vidros”. Caso seja pressionado o botão ‘Remover Camada’ (Figura 4.12-10), é eliminada a última linha da tabela “Camadas do vidro” (Figura 4.12-11).

O botão “Guardar Vidro” (Figura 4.12-12), guarda na variável “projecto” o nome, tipo de vidro e o valor do coeficiente global de transmissão de calor. No caso de ser introduzido o valor do coeficiente global de transmissão de calor, este é guardado directamente. No entanto, se forem introduzidos os valores das variáveis para o cálculo do coeficiente, este é efectuado de acordo com o descrito no subcapítulo 3.2.2. O cálculo do coeficiente global

de transmissão de calor dos vidros é feito através das funções “ug\_simples”, “ug\_duplo” e “ug\_triplo”, respetivamente.

O botão “Ver/Eliminar Vidro” (Figura 4.13-2) disponibiliza os objectos gráficos, de acordo com a Figura 4.13, que permitem ver ou eliminar um vidro. Caso seja pressionado o botão “Ver vidro” (Figura 4.13-6), é enviado para as caixas de texto do “Nome do vidro” (Figura 4.13-3), “Tipo do vidro” (Figura 4.13-4) e “Coeficiente Global de Transmissão de Calor” (Figura 4.13-5) a informação referente ao vidro seleccionado na caixa de listagem “Lista de vidros” (Figura 4.13-8). Para eliminar o vidro deve ser primeiro seleccionado um vidro na caixa de listagem “Lista de vidros” (Figura 4.13-8), só depois deverá ser pressionado o botão “Eliminar vidro” (Figura 4.13-7). Os únicos vidros possíveis de eliminar são os vidros adicionados pelo utilizador.

#### **4.2.5. GUI de caixilho**

O GUI de caixilho (Figura 4.14) permite adicionar novos caixilhos à respectiva estrutura de dados consoante o projeto seleccionado no GUI principal. Também possibilita a visualização dos caixilhos adicionados pelo utilizador ou pré-definidos na estrutura de dados, tal como apenas a eliminação dos elementos adicionados pelo utilizador. Quando accionado o GUI de caixilho, neste só são visíveis os botões: “Adicionar Caixilho” (Figura 4.14-2) e o “Ver/Eliminar Caixilho” (Figura 4.14-3). Consoante o botão seleccionado são visíveis os objetos gráficos de acordo com a Figura 4.14 ou a Figura 4.15, caso se pretenda adicionar o caixilho ou ver/eliminar o mesmo, respetivamente.

Existem duas possibilidades para adicionar um caixilho através do botão “Guardar Caixilho” (Figura 4.14-10). A primeira possibilidade é com recurso à introdução do valor do coeficiente global de transmissão de calor, e a outra possibilidade através da inserção dos valores referentes às caixas de texto 6, 7, 8 e 9 da Figura 4.14. Para efetuar o cálculo de acordo com o referido no subcapítulo 3.2.3. Se for pretendido alternar entre as duas hipóteses de guardar o caixilho, basta seleccionar uma das caixas de verificação, entre o campo 4 e o 5 da Figura 4.14.

Para ver ou eliminar um caixilho, primeiro é necessário seleccionar o botão “Ver/Eliminar Caixilho” (Figura 4.14-3). Seleccionando este botão são visíveis os objectos gráficos de acordo com a Figura 4.15. A informação do nome e do valor do coeficiente global de transmissão de calor do caixilho poderá ser visualizada aquando da selecção do caixilho



na caixa de listagem: “Lista de caixilhos” (Figura 4.15-7), só depois é que deve ser seleccionado o botão “Ver caixilho” (Figura 4.15-5).

Na eliminação de um caixilho, também deverá ser seleccionado primeiro o caixilho na caixa de listagem: “Lista de caixilhos” (Figura 4.15-7), só depois é que deverá ser seleccionado o botão de “Eliminar caixilho” (Figura 4.15-6). Para finalizar a rotina do botão “Eliminar caixilho” (Figura 4.15-6) é actualizada a lista de caixilhos.

Figura 4.14. GUI de adicionar caixilho.

#### Legenda:

- 1 – Caixa de texto: “Nome do caixilho”;
- 2 – Botão: “Adicionar Caixilho”;
- 3 – Botão: “Ver/Eliminar Caixilho”;
- 4 – Caixa de verificação: “Caixilho (método de cálculo geral)”;
- 5 – Caixa de verificação: “Introdução do Coeficiente Global de Transmissão de Calor”;

- 6 – Caixa de texto: “Coeficiente de transmissão de calor bidimensional”;
- 7 – Caixa de texto: “Coeficiente de transmissão de calor”;
- 8 – Caixa de texto: “Largura projectada do caixilho”;
- 9 – Caixa de texto: “Largura visível do painel”;
- 10 – Botão: “Guardar Caixilho”.

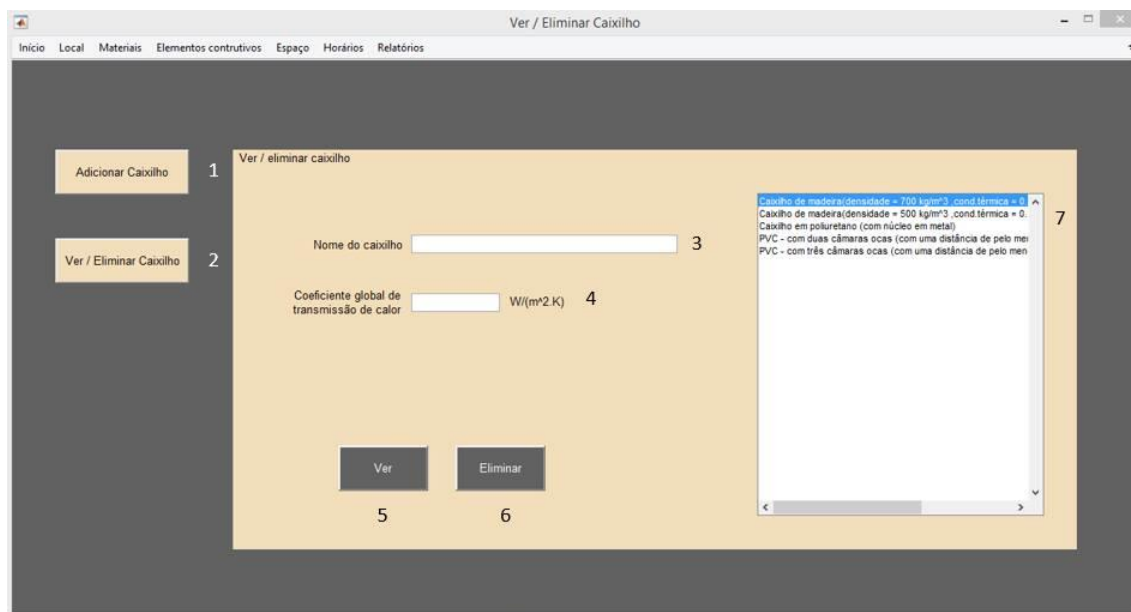


Figura 4.15. GUI de ver / eliminar caixilho.

Legenda:

- |                                                                   |                                              |
|-------------------------------------------------------------------|----------------------------------------------|
| 1 – Botão: “Adicionar Caixilho”;                                  | 5 – Botão: “Ver caixilho”;                   |
| 2 – Botão: “Ver/Eliminar Caixilho”;                               | 6 – Botão: “Eliminar caixilho”;              |
| 3 – Caixa de texto: “Nome do caixilho”;                           | 7 – Caixa de listagem: “Lista de caixilhos”. |
| 4 – Caixa de texto: “Coeficiente global de transmissão de calor”; |                                              |

#### 4.2.6. GUI de adicionar janela simples

O GUI de adicionar janelas simples (Figura 4.16) permite adicionar janelas simples com ou sem painel opaco e dispositivo de oclusão. A Figura 4.16 representa a interface gráfica de introdução de janelas simples da aplicação informática para o cálculo da condução térmica em edifícios.

Quando o GUI de adicionar janelas simples é seleccionado, os campos de vidro e caixilho recebem uma lista com os nomes de vidros e caixilhos que estão disponíveis na estrutura de dados do projecto seleccionado no GUI principal.

Na definição da janela simples é dada a possibilidade de introduzir o valor do coeficiente global de transmissão de calor ou de este ser calculado de acordo com o que foi descrito no subcapítulo 3.2.4.

A selecção do botão “Guardar Janela” (Figura 4.16-2) permite guardar na variável “projecto” informação sobre o nome, tipo (neste caso é janela simples) e o valor do coeficiente global de transmissão de calor da janela. Caso a caixa de verificação “Introduzir Valor do coeficiente global de transmissão de calor” seja seleccionada será sempre guardado o valor introduzido e não o calculado.

Figura 4.16. GUI de adicionar janela simples.

Legenda:

- |                                                                                             |                                                                               |
|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| 1 – Caixa de texto: “Nome da janela”;                                                       | 6 – Painel: “Janela”;                                                         |
| 2 – Botão: “Guardar Janela”;                                                                | 7 – Painel: “Painel Opaco”;                                                   |
| 3 – Caixa de verificação: “Painel opaco”;                                                   | 8 – Painel: “Dispositivo de oclusão”;                                         |
| 4 – Caixa de verificação: “Dispositivo de oclusão”;                                         | 9 – Painel: “Introduzir valor do coeficiente global de transmissão de calor”. |
| 5 – Caixa de verificação: “Introduzir Valor do coeficiente global de transmissão de calor”; |                                                                               |

## 4.2.7. GUI de adicionar janela dupla

O GUI de adicionar janelas duplas (Figura 4.17) permite adicionar janelas duplas introduzindo o valor do coeficiente global de transmissão de calor ou este poderá ser calculado de acordo com o que foi descrito no subcapítulo 3.2.5. A Figura 4.17 representa a interface gráfica de introdução de janelas duplas da aplicação informática para o cálculo da condução térmica em edifícios.

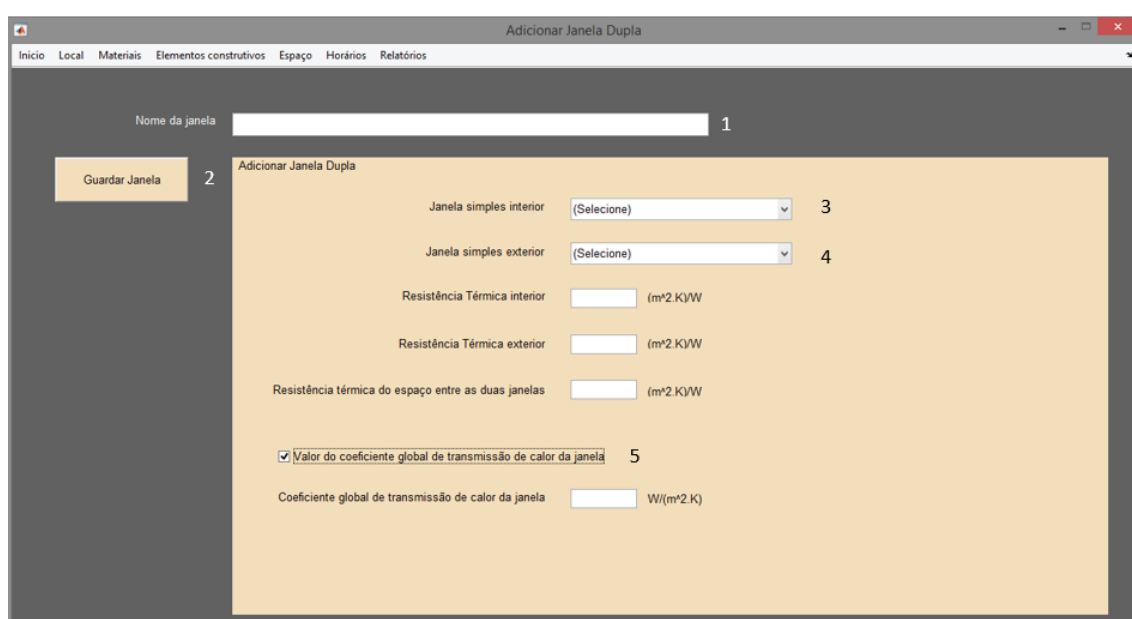


Figura 4.17. GUI de adicionar janela dupla.

Legenda:

- |                                                    |                                                                                                       |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| 1 – Caixa de texto: “Nome da janela”;              | 4 – Menu <i>popup</i> : “Janela simples exterior”;                                                    |
| 2 – Botão: “Guardar Janela”;                       |                                                                                                       |
| 3 – Menu <i>popup</i> : “Janela simples interior”; | 5 – Caixa de verificação: “Introduzir valor do coeficiente global de transmissão de calor da janela”. |

Quando o GUI de adicionar janelas duplas é seleccionado são enviados para os menus *popup* “Janela simples interior” (3) e “Janela simples exterior” (4) a lista com os nomes das janelas simples adicionadas e pré-definidas ao projecto seleccionado no GUI principal.

A informação com o nome da janela, tipo da janela (neste caso janela dupla) e o valor do coeficiente global de transmissão de calor são guardados na variável “projecto” quando é seleccionado o botão “Guardar Janela” (2). Caso a caixa de verificação “Introduzir valor do coeficiente global de transmissão de calor da janela” (4) seja seleccionado, será sempre guardado o valor introduzido e não o calculado.

#### 4.2.8. GUI de adicionar janela de dupla folha no mesmo aro

O GUI de adicionar janela de dupla folha no mesmo aro (Figura 4.18) permite adicionar janelas deste tipo com ou sem painel opaco e dispositivo de oclusão. Dando a possibilidade de introduzir o valor do coeficiente global de transmissão de calor ou de ser calculado de acordo com o que foi descrito no subcapítulo 3.2.6. A Figura 4.18 representa a interface gráfica de introdução de janelas de dupla folha no mesmo aro da aplicação informática para o cálculo da condução térmica em edifícios.

Figura 4.18. GUI de adicionar janela de dupla folha no mesmo aro.

Legenda:

1 – Caixa de texto: “Nome da janela”;

2 – Botão: “Guardar Janela”;

3 – Caixa de verificação: “Painel opaco”;

4 – Caixa de verificação: “Dispositivo de oclusão”;

5 – Caixa de verificação: “Introduzir Valor do coeficiente global de transmissão de calor da janela”;

6 – Painele: “Janela”;

7 – Painele: “Painele Opaco”;

8 – Painele: “Dispositivo de Oclusão”;

9 – Painele: “Introduzir valor do Coeficiente global de transmissão de calor”.

Seleccionando o GUI de adicionar janelas de dupla folha no mesmo aro são carregados os dados nos campos de vidro interior, exterior e caixilho. Com a respectiva nomenclatura de vidros e caixilhos existentes na estrutura de dados do ficheiro “projecto”.

Quando seleccionado o botão “Guardar Janela” (2) a informação sobre o nome da janela, tipo da janela (neste caso é janela de dupla folha no mesmo aro) e o valor do coeficiente global de transmissão de calor é guardada na variável ‘projecto’. Caso a caixa de verificação “Introduzir Valor do coeficiente global de transmissão de calor da janela” (5) esteja seleccionado será sempre guardado o valor introduzido e não o calculado.

#### **4.2.9. GUI de Ver/Eliminar Janela**

O GUI de Ver/Eliminar Janela (Figura 4.19) permite ver ou eliminar as janelas simples, duplas e de dupla folha no mesmo aro. É dada a possibilidade de ver todas as janelas, as pré-definidas e as adicionadas à estrutura de dados. No entanto, só é possível eliminar as janelas adicionadas pelo utilizador. A Figura 4.19 representa a interface gráfica que permite ver ou eliminar janelas da aplicação informática para o cálculo da condução térmica em edifícios.

Para ver a informação do nome, tipo de janela e o valor do coeficiente global de transmissão de calor de uma janela, primeiro deve ser seleccionada a janela na caixa de listagem “Lista de janelas” (6), só depois é que deverá ser pressionado o botão “Ver Janela” (1). A remoção de uma janela da estrutura de dados é feita seleccionando a respectiva janela na caixa de listagem: “Lista de janelas” (6), e pressionando o botão “Eliminar Janela” (2).

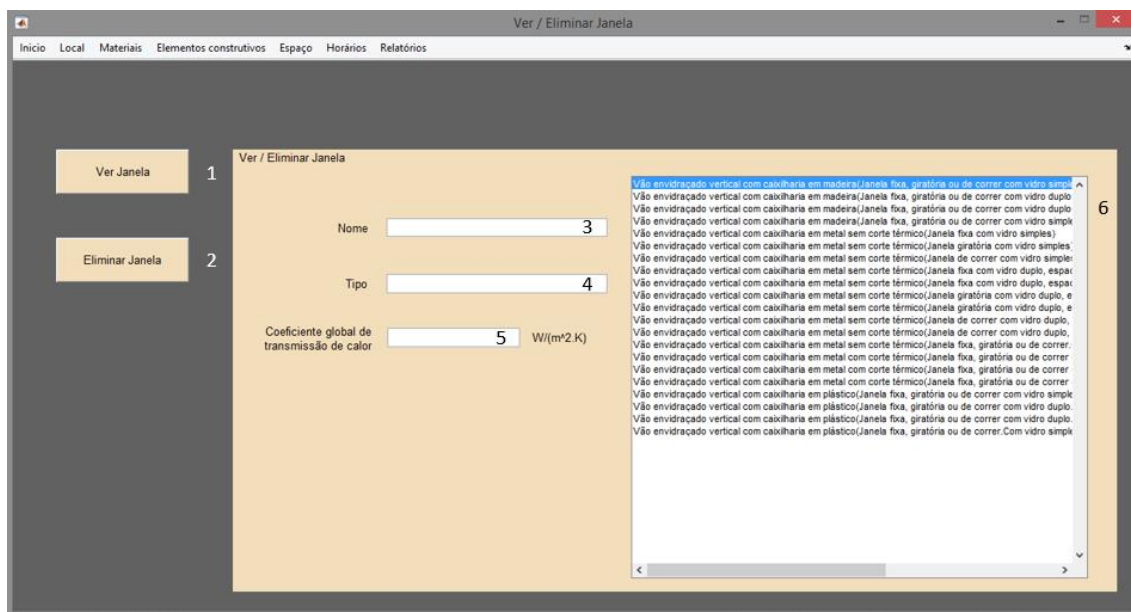


Figura 4.19. GUI de Ver / Eliminar Janela.

Legenda:

- |                                       |                                                                   |
|---------------------------------------|-------------------------------------------------------------------|
| 1 – Botão: “Ver Janela”;              | 5 – Caixa de texto: “Coeficiente global de transmissão de calor”; |
| 2 – Botão: “Eliminar Janela”;         | 6 – Caixa de listagem: “Lista de janelas”.                        |
| 3 – Caixa de texto: “Nome da janela”; |                                                                   |
| 4 – Caixa de texto: “Tipo de janela”; |                                                                   |

#### 4.2.10. GUI de adicionar porta

O GUI de adicionar porta (Figura 4.20) permite adicionar portas com ou sem painel opaco e dispositivo de oclusão. Na definição da porta é dada a possibilidade de introduzir o valor do coeficiente global de transmissão de calor ou este poderá ser calculado de acordo com o que foi descrito no subcapítulo 3.2.8. A Figura 4.20 representa a interface gráfica de introdução de portas da aplicação informática para o cálculo da condução térmica em edifícios.

Quando o GUI de adicionar portas é seleccionado, este envia para os menus *popup* de vidro e caixilho do painel “Porta”, a lista de nomes dos vidros e caixilhos. Esta informação está disponível na estrutura de dados do projecto seleccionado no GUI principal.

O nome da porta e o valor do coeficiente global de transmissão de calor são guardados na variável ‘projecto’ sempre que seleccionado o botão “Guardar porta” (2). Caso a caixa de

verificação “Introduzir Valor do coeficiente global de transmissão de calor da porta” (4) esteja seleccionado, será guardado o valor introduzido em vez do calculado de acordo com o subcapítulo 3.2.8.

Figura 4.20. GUI de adicionar porta.

Legenda:

- |                                                                                                      |                                                                                        |
|------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| 1 – Caixa de texto: “Nome porta”;                                                                    | 5 – Painel: “Porta”;                                                                   |
| 2 – Botão: “Guardar porta”;                                                                          | 6 – Painel: “Painel opaco”;                                                            |
| 3 – Caixa de verificação: “Painel opaco”;                                                            | 7 – Painel: “Introduzir valor do coeficiente global de transmissão de calor da porta”. |
| 4 – Caixa de verificação: “Introduzir Valor do coeficiente global de transmissão de calor da porta”; |                                                                                        |

#### 4.2.11.GUI de Ver/Eliminar Porta

O GUI de Ver/Eliminar Porta (Figura 4.21) permite ver todas as portas pré-definidas e as adicionadas à estrutura de dados. Na eliminação de portas, só é possível eliminar as portas adicionadas pelo utilizador. A Figura 4.21 representa a interface gráfica que permite ver ou eliminar portas da aplicação informática para o cálculo da condução térmica em edifícios.



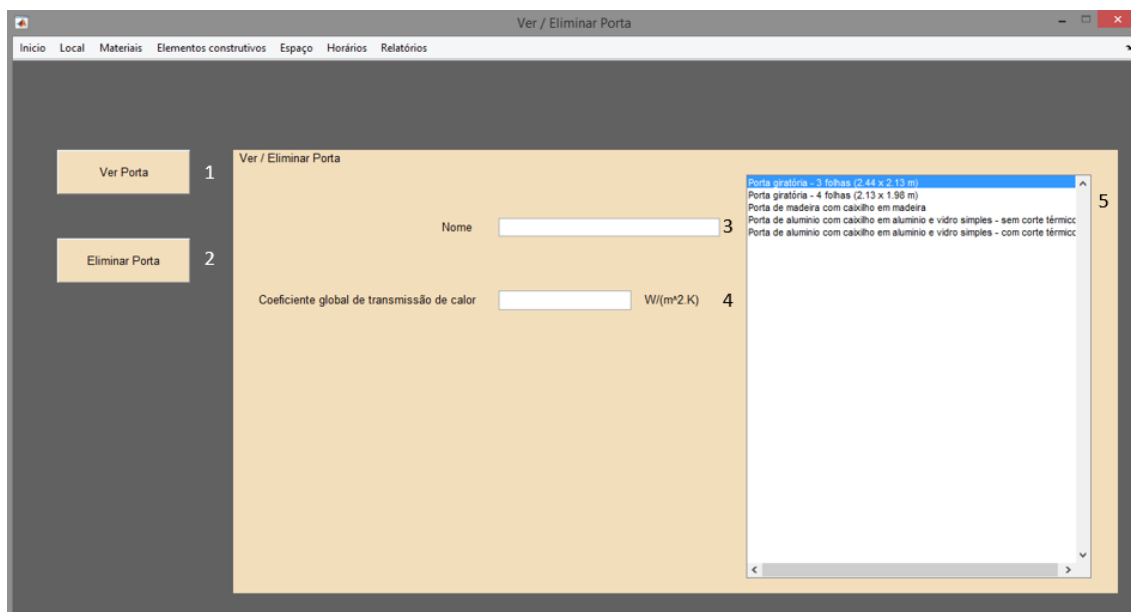


Figura 4.21. GUI de ver / eliminar porta.

Legenda:

- |                                      |                                                                            |
|--------------------------------------|----------------------------------------------------------------------------|
| 1 – Botão: “Ver Porta”;              | 4 – Caixa de texto: “Coeficiente global de transmissão de calor da porta”; |
| 2 – Botão: “Eliminar Porta”;         | 5 – Caixa de listagem: “Lista de portas”.                                  |
| 3 – Caixa de texto: “Nome da porta”; |                                                                            |

Para ver a informação do nome e o valor do coeficiente global de transmissão de calor de uma porta, primeiro deverá ser seleccionada a porta na caixa de listagem: “Lista de portas” (5). Só depois é que deverá ser escolhido o botão “Ver Porta” (1). Caso se pretenda eliminar a porta é necessário seleccionar a porta a remover da estrutura de dados na caixa de listagem: “Lista de portas” (5). De seguida deverá ser pressionado o botão “Eliminar janela” (2).

#### 4.2.12.GUI de adicionar parede, tecto, cobertura e pavimento

O GUI de adicionar parede, tecto, cobertura e pavimento (Figura 4.22) permite adicionar o elemento construtivo escolhido, no menu *popup* “Tipo de elemento construtivo” (2), à estrutura de dados do projecto, no GUI principal. A Figura 4.22 representa a interface gráfica que permite adicionar paredes, tectos, coberturas e pavimentos na aplicação informática.

Figura 4.22. GUI de adicionar parede, tecto, cobertura e pavimento.

#### Legenda:

- |                                                         |                                                                                                               |
|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| 1 – Caixa de texto: “Nome do elemento construtivo”;     | 9 – Caixa de texto: “Resistência Térmica interior”;                                                           |
| 2 – Menu <i>popup</i> : “Tipo de elemento construtivo”; | 10 – Caixa de texto: “Resistência Térmica exterior”;                                                          |
| 3 – Menu <i>popup</i> : “Tipo de material”;             | 11 – Caixa de verificação: “Introduzir o coeficiente global de transmissão de calor do elemento construtivo”; |
| 4 – Menu <i>popup</i> : “Material”;                     | 12 – Botão: “Adicionar camada”;                                                                               |
| 5 – Caixa de texto: “Massa volúmica”;                   | 13 – Botão: “Remover camada”;                                                                                 |
| 6 – Caixa de texto: “Condutibilidade térmica”;          | 14 – Botão: “Guardar elemento construtivo”.                                                                   |
| 7 – Caixa de texto: “Espessura”;                        |                                                                                                               |
| 8 – Tabela: “Constituição do elemento construtivo”;     |                                                                                                               |

Ao pressionar o botão “Guardar elemento construtivo” (14) é guardada a informação relativa: ao nome, tipo, informação contida na tabela da constituição do elemento construtivo, assim como a resistência térmica interior e exterior. Caso a caixa de verificação “Introduzir o coeficiente global de transmissão de calor do elemento construtivo” seja seleccionada é apenas guardado o nome do material, tipo de elemento construtivo e o valor de coeficiente global de transmissão de calor.

Para adicionar uma linha com informação na tabela “Constituição do elemento construtivo” (8) é necessário seleccionar o botão “Adicionar camada” (12). Contudo, é necessário previamente definir a informação a introduzir na tabela. Escolhendo o tipo de material no menu *popup* (3) da Figura 4.22 é enviada a lista com os nomes dos materiais dessa classe para o menu *popup* “Material” (4). Seleccionando o material, são enviados os valores da massa volúmica e condutibilidade térmica para as caixas de texto 5 e 6 da Figura 4.22. A função “tipo\_material” recebe informação do tipo de material seleccionado e cria a lista de nomes da classe de material. Para completar a informação a enviar para uma linha da tabela, deverá ser introduzida a espessura da camada. As funções que enviam a informação sobre a massa volúmica e a condutibilidade térmica são feitas através da função “rho”. A função “material\_nome” permite estabelecer o material seleccionado no menu *popup* “Material”. Caso seja seleccionado o botão “Remover camada” (13), é eliminada a última linha da tabela da constituição do elemento construtivo.

#### **4.2.13. GUI de Ver/Eliminar Parede, Tecto, Cobertura e Pavimento**

O GUI Ver/Eliminar Parede, Tecto, Cobertura e Pavimento (Figura 4.23) permite ver todos os elementos construtivos pré-definidos e os adicionados à estrutura de dados. No entanto, apenas os elementos adicionados pelo utilizador poderão ser eliminados posteriormente. A Figura 4.23 representa a interface gráfica que permite ver ou eliminar paredes, tectos, coberturas e pavimentos da aplicação informática para o cálculo da condução térmica em edifícios.

Para ver a informação: nome, tipo de elemento construtivo, tabela da constituição do elemento construtivo, resistência térmica interior, resistência térmica exterior e o valor do coeficiente global de transmissão de calor do elemento construtivo, deverá ser seleccionado o elemento construtivo na caixa de listagem “Lista de elementos construtivos” (9). Só depois deverá ser pressionado o botão “Ver elemento construtivo” (7). Caso se pretenda eliminar o elemento construtivo também deverá ser primeiramente seleccionado este na respectiva lista, só depois deverá ser pressionado o botão “Eliminar elemento construtivo” (8).

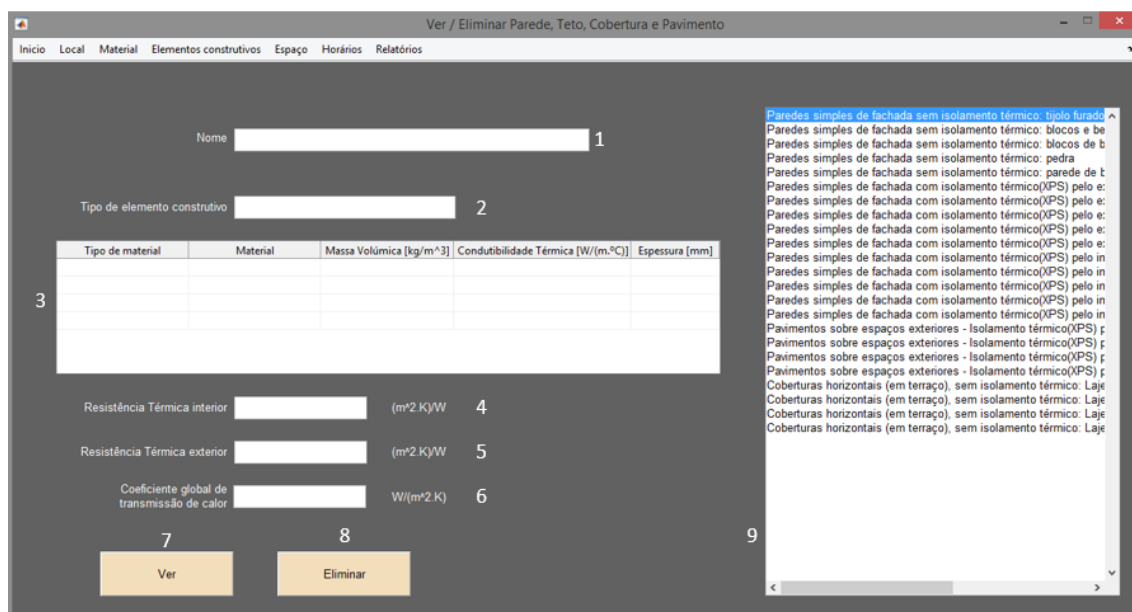


Figura 4.23. GUI de Ver / Eliminar Parede, Tecto, Cobertura e Pavimento.

#### Legenda:

1 – Caixa de texto: “Nome do elemento construtivo”;

2 – Caixa de texto: “Tipo de elemento construtivo”;

3 – Tabela: “Constituição do elemento construtivo”;

4 – Caixa de texto: “Resistência Térmica interior”;

5 – Caixa de texto: “Resistência Térmica exterior”;

6 – Caixa de texto: “Introduzir Coeficiente global de transmissão de calor do elemento construtivo”;

7 – Botão: “Ver elemento construtivo”;

8 – Botão: “Eliminar elemento construtivo”;

9 – Caixa de listagem: “Lista de elementos construtivos”.

#### 4.2.14.GUI de Adicionar Espaço

O GUI de Adicionar Espaço (Figura 4.24) permite ao utilizador criar espaços onde irá ser feita uma análise climática. A Figura 4.24 representa a interface gráfica que permite adicionar um espaço na aplicação informática para o cálculo da condução térmica em edifícios.

Figura 4.24. GUI de adicionar espaço.

#### Legenda:

- 1 – Caixa de texto: “Nome do espaço”;
- 2 – Caixa de texto: “Temperatura bolbo seco interior”;
- 3 – Caixa de texto: “Humidade relativa interior”;
- 4 – Caixa de texto: “Área do espaço”;
- 5 – Botão: “Guardar espaço”;
- 6 – Painel: “Ganhos de calor externos”;
- 7 – Painel: “Ganhos de calor internos”;
- 8 – Caixa de verificação: “Verificação do elemento referente ao ganho de calor externo”;
- 9 – Menu *popup*: “Tipo de elemento referente ao ganho de calor externo”;
- 10 – Menu *popup*: “Elemento referente ao ganho de calor externo”;
- 11 – Caixa de texto: “Quantidade de um dado elemento referente ao ganho de calor externo”;
- 12 – Caixa de texto: “Área de um dado elemento referente ao ganho de calor externo”;
- 13 – Caixa de verificação: “Verificação do elemento referente ao ganho de calor interno”;
- 14 – Menu *popup*: “Tipo de elemento referente ao ganho de calor interno”;
- 15 – Menu *popup*: “Elemento referente ao ganho de calor interno”;
- 16 – Caixa de texto: “Quantidade de um dado elemento referente ao ganho de calor interno”;
- 17 – Caixa de texto: “Área de um dado elemento referente ao ganho de calor externo”;
- 18 – Caixa de texto: “Temperatura do espaço adjacente de um dado elemento referente ao ganho de calor externo”;
- 19 – Painel: “Ocupação”;
- 20 – Painel: “Cargas diversas”;
- 21 – Painel: “Infiltrações”;
- 22 – Painel: “Equipamentos eléctricos”;
- 23 – Painel: “Iluminação”.

Esta interface gráfica permite obter informação sobre: o nome do espaço, temperatura interior, humidade relativa interior, área do espaço, informações sobre a envolvente exterior contabilizada nos ganhos de calor externos, informação sobre a envolvente interior nos ganhos de calor internos, informação sobre a ocupação, iluminação, infiltração de ar e de cargas diversas. A informação referente aos elementos da envolvente exterior consiste no tipo de elemento, nome do elemento, a sua quantidade e área. Na envolvente interior é pedida uma informação adicional em relação à envolvente exterior, a temperatura do espaço adjacente. No painel “Ganhos de calor externos” (6) não é pedida nenhuma temperatura visto se referir à temperatura exterior. Para definir o nome, tipo e coeficiente global de transmissão de calor da envolvente opaca interior e exterior é utilizada a função “elementos\_espaco”.

No painel “Ocupação” (19) existe a possibilidade de introduzir o número de ocupantes do espaço ou de introduzir o valor da densidade de ocupação. Nas outras caixas de texto do painel “Ocupação” (19) são introduzidos os ganhos de calor latente e sensível por unidade de área do espaço.

O painel “Cargas diversas” (20), possibilita introduzir as cargas térmicas sensíveis ou latentes que não se enquadrem nas cargas térmicas devido à ocupação, iluminação, equipamentos eléctricos ou infiltração de ar. No painel “Equipamentos eléctricos” (22) é possível introduzir o valor dos ganhos de calor através dos equipamentos eléctricos ou da sua densidade de utilização. O painel “Infiltrações” (21) requer o valor do caudal volumico de ar, valores de humidade relativa exterior de projecto, tanto de Verão como de Inverno. No painel “Iluminação” é possível introduzir o valor dos ganhos de calor através da iluminação ou introduzir a densidade de iluminação do espaço.

#### **4.2.15.GUI de Ver/Eliminar Espaço**

O GUI Ver/Eliminar Espaço (Figura 4.25) permite ao utilizador consultar e apagar os espaços adicionados no GUI de Adicionar Espaço. A Figura 4.25 representa a interface gráfica que permite ver ou eliminar um espaço da aplicação informática para o cálculo da condução térmica em edifícios.

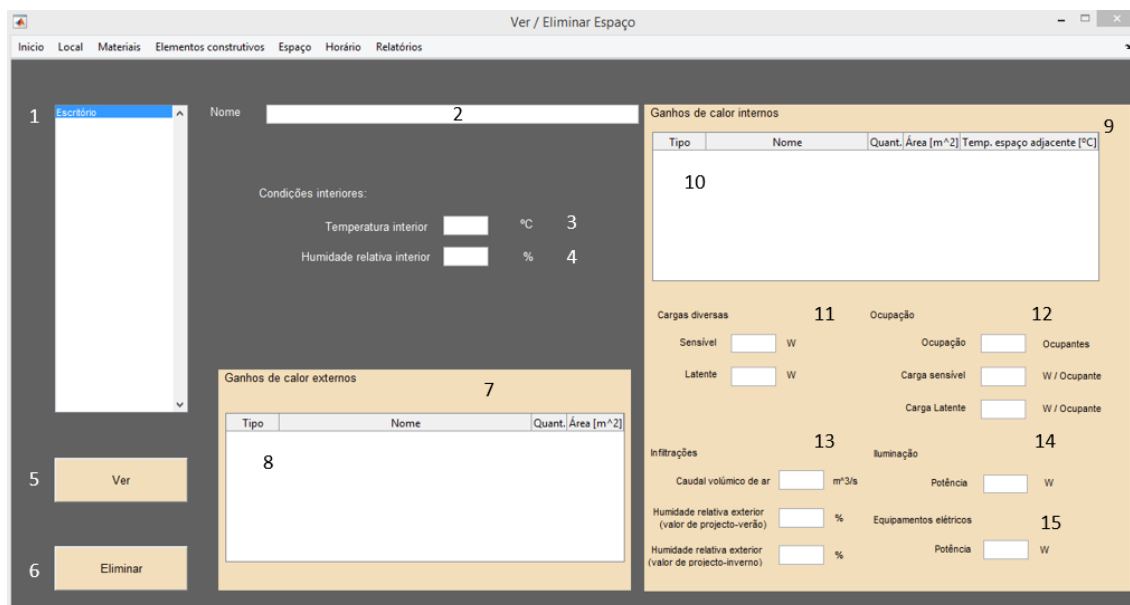


Figura 4.25. GUI de ver / eliminar espaço.

#### Legenda:

- 1 – Caixa de listagem: “Lista de espaços”;
- 2 – Caixa de texto: “Nome do espaço”;
- 3 – Caixa de texto: “Temperatura interior”;
- 4 – Caixa de texto: “Humidade relativa interior”;
- 5 – Botão: “Ver espaço”;
- 6 – Botão: “Eliminar espaço”;
- 7 – Painel: “Ganhos de calor externos”;
- 8 – Tabela: “Elementos referentes a ganhos externos”;
- 9 – Painel: “Ganhos de calor internos”;
- 10 – Tabela: “Elementos referentes a ganhos internos”;
- 11 – Painel: “Cargas diversas”;
- 12 – Painel: “Ocupação”;
- 13 – Painel: “Infiltrações”;
- 14 – Painel: “Iluminação”;
- 15 – Painel: “Equipamentos eléctricos”.

Caso seja accionado o botão “Ver espaço” (5), e tenha sido seleccionada um espaço na caixa de listagem: “Lista de espaços” (1), é enviada a informação referente ao espaço seleccionado, para os campos: “Nome do espaço” (2), “Temperatura interior” (3), “Humidade relativa interior” (4), “Elementos referentes a ganhos externos” (8), “Elementos referentes a ganhos internos” (10), campos do painel “Cargas diversas” (11),

campos do painel “Ocupação” (12), campos do painel “Iluminação” (14), campos do painel “Equipamentos eléctricos” (15) e para o campo do painel “Infiltrações” (13).

Para eliminar um espaço deverá ser primeiramente seleccionado o espaço que se pretende eliminar na caixa de listagem “Lista de espaços” (1), só depois é que deverá ser accionado o botão “Eliminar espaço” (6).

#### 4.2.16.GUI de Adicionar Horário

O GUI Adicionar Horário (Figura 4.26) permite ao utilizador adicionar, ver ou eliminar um horário. A Figura 4.26 representa a interface gráfica que permite adicionar, ver ou eliminar horários na aplicação informática para o cálculo da condução térmica em edifícios.

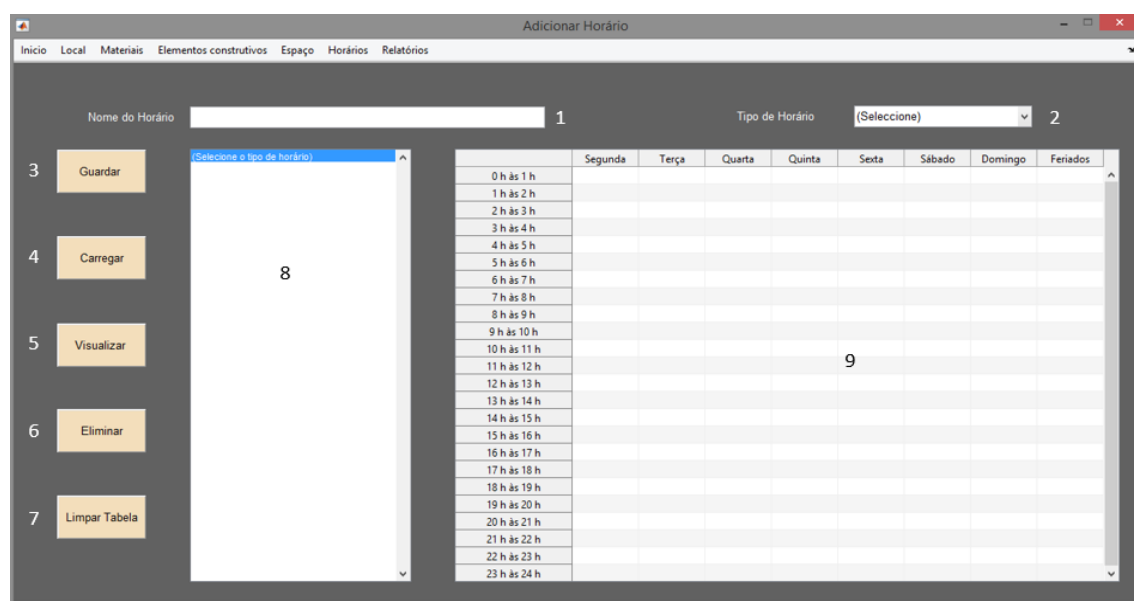


Figura 4.26. GUI de adicionar horário.

Legenda:

- |                                            |                                            |
|--------------------------------------------|--------------------------------------------|
| 1 – Caixa de texto: “Nome do Horário”;     | 6 – Botão: “Eliminar horário”;             |
| 2 – Menu <i>popup</i> : “Tipo de Horário”; | 7 – Botão: “Limpar Tabela”;                |
| 3 – Botão: “Guardar horário”;              | 8 – Caixa de listagem: “Lista de horário”; |
| 4 – Botão: “Carregar horário”;             | 9 – Tabela: “Coeficientes do horário”.     |
| 5 – Botão: “Visualizar horário”;           |                                            |



Para adicionar um novo horário existe a possibilidade de introduzir os valores na tabela “Coeficientes do horário” (9), ou carregar um ficheiro Excel com os coeficientes dos padrões de referência de utilização dos edifícios, através do botão “Carregar horário” (4). De notar que estes mesmos coeficientes são representados em percentagem. A informação guardada na variável ‘projecto’ desta interface gráfica é o nome e tipo de horário, tal como a tabela com os coeficientes do horário.

Para visualizar um horário, primeiro deverá ser seleccionado o tipo de horário no menu *popup* “Tipo de Horário” (2). Com esta acção é actualizada a caixa de listagem “Lista de horário” (8) com a lista dos nomes dos horários do tipo escolhido. Estando seleccionado o tipo e aquando da selecção de um horário, através do botão “Visualizar horário” (5), é visível o nome do horário no campo 1 e os coeficientes do horário no campo 9. Para eliminar um horário, o procedimento é semelhante ao de visualizar, o que diferencia a selecção do botão “Visualizar horário” (5) que é substituído pelo botão “Eliminar horário” (6). Para facilitar a limpeza da tabela com os coeficientes do horário, a interface gráfica é dotada do botão “Limpar Tabela” (7).

#### **4.2.17.GUI de Definir Horário**

O GUI Definir Horário (Figura 4.27) é a interface gráfica onde são definidos os horários de um espaço. A Figura 4.27 representa a interface gráfica que permite definir o horário de um espaço da aplicação informática para o cálculo da condução térmica em edifícios.

Quando a interface é accionada é enviado para o menu *popup* “Espaço” (1) uma listagem contendo todos os nomes dos espaços criados no GUI de Adicionar Espaço. São também enviados os nomes dos horários para os menus *popup* numerados dos painéis: “Horário Ocupação” (2), “Horário Equipamentos eléctricos” (3), “Horário Iluminação” (4) e “Horário Cargas diversas” (5). Os menus *popup* numerados e as tabelas dos diversos painéis possibilitam diferenciar os horários de um tipo de carga térmica interna ao longo do ano.

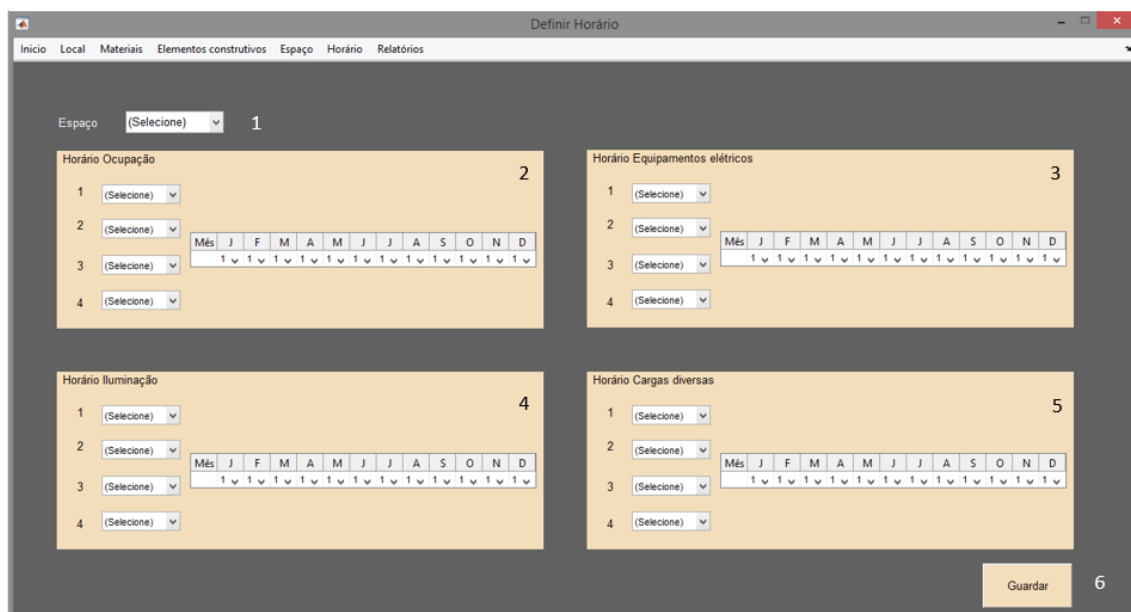


Figura 4.27. GUI de definir horário

Legenda:

- 1 – Menu *popup*: “Espaço”;
- 2 – Paine: “Horário Ocupação”;
- 3 – Paine: “Horário Equipamentos eléctricos”;
- 4 – Paine: “Horário Iluminação”;
- 5 – Paine: “Horário Cargas diversas”;
- 6 – Botão: “Guardar”.

#### 4.2.18. GUI de Relatórios

O GUI Relatórios (Figura 4.28) é a interface gráfica onde é feito o cálculo das cargas térmicas. A Figura 4.28 representa a interface gráfica que permite criar relatórios da aplicação informática para o cálculo da condução térmica em edifícios.

As cargas térmicas são calculadas para as condições interiores inseridas no GUI “Adicionar Espaço” e para as obtidas a partir do ficheiro climático adicionado no GUI “Local”. Ou seja, é feita a análise horária das cargas térmicas. Para a análise horária das cargas térmicas é necessário definir o dia da semana referente a 1 de Janeiro, através do menu *popup* “Dia da semana referente a 1 de Janeiro” (9) da interface gráfica. O campo “Ver um dia da simulação” (11) permite ao utilizador fazer um filtro na análise horária das cargas térmicas.

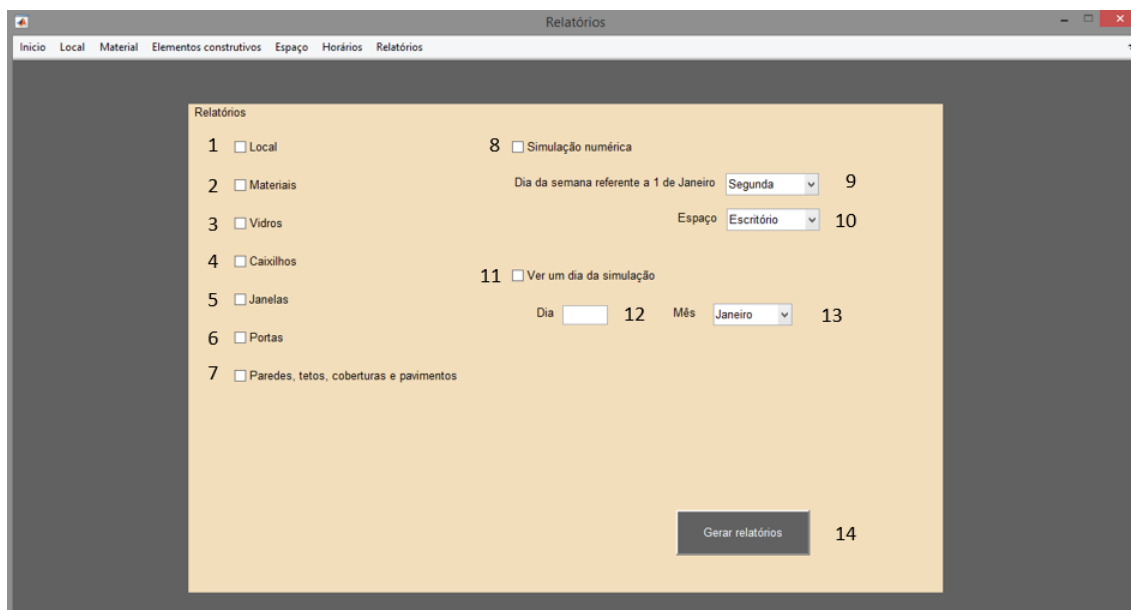


Figura 4.28. GUI de relatórios.

#### Legenda:

- |                                                                      |                                                                  |
|----------------------------------------------------------------------|------------------------------------------------------------------|
| 1 – Caixa de verificação: “Local”;                                   | 9 – Menu <i>popup</i> : “Dia da semana referente a 1 e Janeiro”; |
| 2 – Caixa de verificação: “Materiais”;                               | 10 – Menu <i>popup</i> : “Espaço”;                               |
| 3 – Caixa de verificação: “Vidros”;                                  | 11 – Caixa de verificação: “Ver um dia da simulação”;            |
| 4 – Caixa de verificação: “Caixilhos”;                               | 12 – Caixa de texto: “Dia”;                                      |
| 5 – Caixa de verificação: “Janelas”;                                 | 13 – Menu <i>popup</i> : “Mês”;                                  |
| 6 – Caixa de verificação: “Portas”;                                  | 14 – Botão: “Gerar relatórios”.                                  |
| 7 – Caixa de verificação: “Paredes, tetos, coberturas e pavimentos”; |                                                                  |
| 8 – Caixa de verificação: “Simulação numérica”;                      |                                                                  |

Além do cálculo das cargas térmicas nesta interface gráfica, também é permitido exportar para Excel o conteúdo das estruturas de dados de: “Local”, “Materiais”, “Vidros”, “Caixilhos”, “Janelas”, “Portas”, “Paredes”, “Tectos”, “Coberturas” e “Pavimentos”. Os relatórios que são gerados são exportados para o ficheiro em Excel denominado de “Relatórios”. Este ficheiro é gerado automaticamente, e é guardado no directório de instalação da aplicação. Os resultados do cálculo das cargas térmicas de aquecimento e arrefecimento estão disponíveis na folha “Análise de projecto”. A análise anual é disponibilizada na folha “Análise Anual” do ficheiro “Relatórios”. Na exportação da

análise de projecto (Anexo G) é disponibilizada a informação sobre cargas térmicas devido a: envolvente externa, envolvente interna, ocupação cargas diversas (sensível e latente), iluminação, equipamentos eléctricos, infiltrações de ar (sensível e latente), cargas térmicas de arrefecimento (sensível, latente e total) e cargas térmicas de aquecimento (sensível, latente e total). No caso da análise anual (Anexo H) para além dos parâmetros descritos na análise de projecto, a aplicação ainda exporta informação sobre: dia do ano (com valores de 1 a 365), hora do dia (com valores de 1 a 24), coeficientes dos horários de ocupação, iluminação, equipamentos eléctricos e de cargas diversas.

As funções utilizadas neste ambiente gráfico são: “myinterp1”, “altitude\_pressao”, “def\_horarios”, “calc\_entalpia”, “analise\_dia”, “di\_semana”, “coef\_horarios”, “analise\_hora”, “hora\_projeto”.

A função “myinterp1” é a função que faz a interpolação da tabela de vapor. Esta função é utilizada na função “calc\_entalpia”, permitindo calcular a entalpia do ar de acordo com o que foi descrito no subcapítulo 3.4.3. “Calc\_entalpia” é utilizada na função “hora\_projeto” e na “analise\_hora”.

A “Altitude\_pressao” é a função que determina a pressão atmosférica com o efeito de altitude, sendo esta função utilizada na função de “hora\_projeto”. A função “hora\_projeto” é a função que permite calcular as cargas térmicas de projecto, isto é, com as condições exteriores definidas no GUI de local. Esta função é utilizada no GUI de relatórios.

Para definir o horário de ocupação, iluminação, equipamentos, cargas diversas e equipamentos eléctricos em cada mês do ano é utilizada a função “def\_horarios”, sendo executada na função “analise\_hora”.

A função “di\_semana” é a função que define o dia da semana ao longo do ano e os feriados nacionais que têm data fixa. Esta função é utilizada na função “analise\_hora”.

Para fazer a ligação entre os coeficientes de um determinado horário e o dia da semana ou feriado, é usada a função “coef\_horarios”, sendo esta executada na função “analise\_hora”.

A função “analise\_hora” é a função que faz o cálculo das cargas térmicas ao longo de todo o ano. Esta função é utilizada no GUI de relatórios.

Para definir a primeira linha do dia escolhido para o filtro da simulação, sendo esta função executada no GUI de relatórios e só sendo executada caso seja feito um filtro na simulação numérica, é utilizada a função “analise\_dia”.



## 5. Validação da aplicação

Este capítulo tem como objectivo a validação da aplicação desenvolvida para o cálculo da condução térmica em edifícios. Esta validação foi realizada com recurso ao programa comercial Hourly Analysis Program (HAP), desenvolvido pela empresa de AVAC Carrier. Também foram efectuados cálculos analíticos, utilizando a mesma metodologia de cálculo adoptada na aplicação desenvolvida. Esta última análise foi realizada com o intuito de identificar possíveis erros na programação efectuada.

A análise foi realizada considerando como caso de estudo uma porção de um edifício com diferentes elementos construtivos e materiais. Este é rigorosamente igual nas análises efectuadas, como também não foi escolhido com base em nenhuma construção pré-existente.

### 5.1. Caso de estudo

Para se parametrizar o caso de estudo definiu-se que este se encontra na cidade de Lisboa. Foram consideradas como condições exteriores de Verão, uma temperatura de bolbo seco igual a 33 °C e humidade relativa igual a 40%. As condições exteriores de Inverno consideradas foram uma temperatura de bolbo seco igual a 5 °C e humidade relativa igual a 90%. Como amplitude térmica diária foi considerado o valor de 10 °C.

A utilização do espaço foi definida como escritórios com uma área de 25 m<sup>2</sup>. As condições interiores do espaço são consideradas como constantes ao longo do ano. O valor da temperatura de bolbo seco interior foi definido para 22 °C com uma humidade relativa igual a 50%. O espaço definido como caso de estudo é representado em planta na Figura 5.1. Nessa mesma representação são definidos os componentes constituintes do espaço a ser alvo de estudo.

A envolvente externa do local é constituída por duas paredes, dois vãos envidraçados, cobertura e pavimento. As paredes exteriores, cada uma, têm uma área de 15 m<sup>2</sup>. A parede exterior 1 (Figura 5.1-1) tem um vão envidraçado vertical com caixilharia em madeira (janela fixa, giratória ou de correr com vidro simples) (Figura 5.1-2) com uma área igual a 4 m<sup>2</sup> e um coeficiente global de transmissão de calor igual a 5,1 W/m<sup>2</sup>.K. A Tabela 5.1 contém a informação da constituição da parede exterior 1.

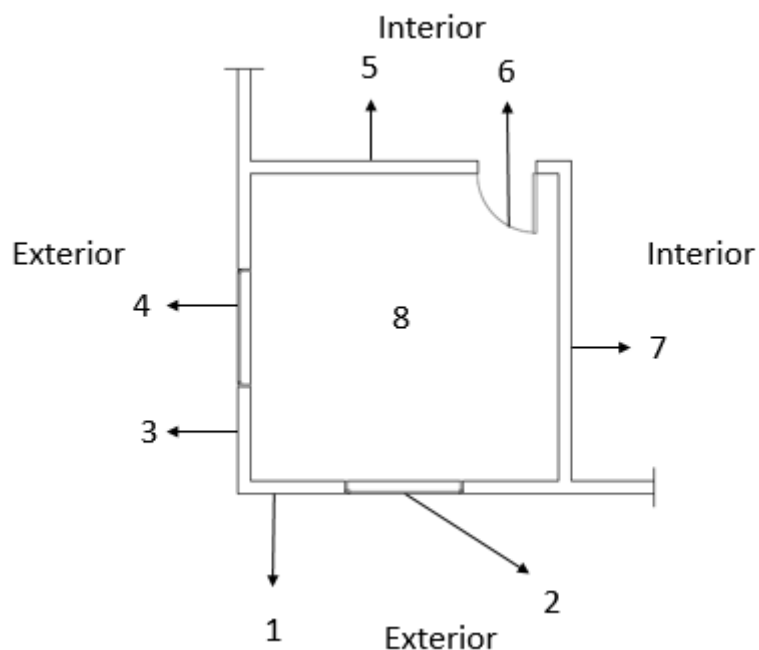


Figura 5.1. Representação em planta do espaço definido como caso de estudo.

Legenda:

- |                                           |                             |
|-------------------------------------------|-----------------------------|
| 1 – Parede exterior 1;                    | 5 – Parede interior 1;      |
| 2 – Vão envidraçado da parede exterior 1; | 6 – Porta;                  |
| 3 – Parede exterior 2;                    | 7 – Parede interior 2;      |
| 4 – Vão envidraçado da parede exterior 2; | 8 – Espaço (Caso de estudo) |

Tabela 5.1. Parede exterior 1.

Elemento de Camada	e (m)	$\lambda$ (W/m.K)	R (m <sup>2</sup> .K/W)	U (W/m <sup>2</sup> .K)	Massa Volúmica (kg/m <sup>3</sup> )	Calor Específico (kJ/(kg.k))
Resistência Interior	-	-	0,130	<b>0,603</b>	-	1,007
Reboco Hidrófugo	0,025	1,30	0,019		1800	1,085
Tijolo de 7 cm	0,070	-	0,190		-	0,835
Caixa de ar	0,040	-	0,180		-	1,007
Isolamento em Poliestireno Extrudido	0,030	0,037	0,811		30	1,210
Tijolo de 11 cm	0,110	-	0,270		-	0,835
Reboco hidrófugo	0,025	1,30	0,019		1800	1,085
Resistência Exterior	-	-	0,040		-	1,007



A parede exterior 2 (Figura 5.1-3) tem um vão envidraçado vertical com caixilharia em madeira, janela fixa, giratória ou de correr com vidro simples em cada janela, com uma distância entre janelas de 50 a 100 mm) (Figura 5.1-4) com uma área igual a 4 m<sup>2</sup> e um coeficiente global de transmissão de calor igual a 2,5 W/m<sup>2</sup>.K. A Tabela 5.2 contém a informação da constituição da parede exterior 2.

Tabela 5.2. Parede exterior 2.

Elemento de Camada	e (m)	$\lambda$ (W/m.K)	R (m <sup>2</sup> .K/W)	U (W/m <sup>2</sup> .K)	Massa Volúmica (kg/m <sup>3</sup> )	Calor Específico (kJ/(kg.k))
Resistência Interior	-	-	0,130	<b>0,257</b>	-	1,007
Painel Sandwich (XPS)	0,080	-	2,250		31	1,100
Tijolo de 7 cm	0,070	-	0,190		-	0,835
Caixa de ar	0,040	-	0,180		-	1,007
Isolamento em Poliestireno Extrudido	0,030	0,037	0,811		30	1,210
Tijolo de 11 cm	0,110	-	0,270		-	0,835
Reboco hidrófugo	0,025	1,30	0,019		1800	1,085
Resistência Exterior	-	-	0,040		-	1,007

O pavimento é o pavimento da estrutura de dados de elementos construtivos com o nome “Pavimento sobre espaço exterior, Laje aligeirada, blocos cerâmicos”. Com uma área igual a 25 m<sup>2</sup> e um coeficiente global de transmissão de calor igual a 0,81 W/m<sup>2</sup>.K.

Na cobertura do espaço foi considerado que esta está em contacto com o exterior e tem uma área igual a 25 m<sup>2</sup>. A Tabela 5.3 contém a informação da constituição da cobertura do espaço.

A envolvente interna do local é constituída por duas paredes e uma porta. Foi considerado que a temperatura do espaço adjacente dos elementos da envolvente interna é igual a 20 °C. As paredes interiores têm uma área igual a 15 m<sup>2</sup>. Sendo que a parede interior 1 (Figura 5.1-5) tem uma porta (Figura 5.1-6) com uma área igual a 1,995 m<sup>2</sup> e um coeficiente global de transmissão de calor igual a 2,61 W/m<sup>2</sup>.K. A Tabela 5.4 contém a informação da constituição da parede interior 1. A Tabela 5.5 contém a informação da constituição da parede interior 2.

Tabela 5.3. Cobertura do espaço.

Elemento de Camada	e (m)	$\lambda$ (W/m.K)	R (m <sup>2</sup> .K/W)	U (W/m <sup>2</sup> .K)	Massa Volúmica (kg/m <sup>3</sup> )	Calor Específico (kJ/(kg.k))
Resistência Exterior	-	-	0,040	<b>0,411</b>	-	1,007
Laje Flutuante em Betão Armado	0,020	2,300	0,009		2300	0,750
Camada separadora/geotêxtil	0,005	0,230	0,022		1000	0,920
Isolamento Térmico - RoofMate	0,025	0,035	0,714		35	1,400
Tela asfáltica	0,025	0,230	0,109		-	0,920
Camada de forma em betão leve de argila expandida	0,010	0,850	0,012		1500	0,780
Betão Celular autoclavado	0,205	0,160	1,281		450	0,750
Betão Armado	0,180	2,300	0,078		2300	0,750
Resistência Interior	-	-	0,170		-	1,007

Tabela 5.4. Parede interior 1.

Elemento de Camada	e (m)	$\lambda$ (W/m.K)	R (m <sup>2</sup> .K/W)	U (W/m <sup>2</sup> .K)	Massa Volúmica (kg/m <sup>3</sup> )	Calor Específico (kJ/(kg.k))
Resistência Interior	-	-	0,130	<b>1,453</b>	-	1,007
Reboco Hidrófugo	0,025	1,30	0,019		1800	1,085
Tijolo de 15 cm	0,150	-	0,390		-	0,835
Reboco hidrófugo	0,025	1,30	0,019		1800	1,085
Resistência Interior	-	-	0,130		-	1,007

Tabela 5.5. Parede interior 2.

Elemento de Camada	e (m)	$\lambda$ (W/m.K)	R (m <sup>2</sup> .K/W)	U (W/m <sup>2</sup> .K)	Massa Volúmica (kg/m <sup>3</sup> )	Calor Específico (kJ/(kg.k))
Resistência Interior	-	-	0,130	<b>0,352</b>	-	1,007
Painel Sandwich (XPS)	0,040	-	1,200		31	1,100
Tijolo de 7 cm	0,070	-	0,190		-	0,835
Caixa de ar	0,040	-	0,180		-	1,007
Isolamento em Poliestireno Extrudido	0,030	0,037	0,811		30	1,210
Tijolo de 11 cm	0,110	-	0,270		-	0,835
Reboco hidrófugo	0,025	1,30	0,019		1800	1,085
Resistência Interior	-	-	0,040		-	1,007

Para a densidade de ocupação do espaço foi considerado o valor de 15 m<sup>2</sup> por ocupante, libertando cada pessoa uma carga sensível igual a 75 W e uma carga latente de 55 W. Relativamente à iluminação foi considerada uma densidade de iluminação igual a 10,5 W/m<sup>2</sup>. Em relação aos equipamentos eléctricos foi admitido um valor de 15 W/m<sup>2</sup> como densidade de calor libertado. Quanto às cargas térmicas diversas, não foram consideradas cargas sensíveis nem latentes adicionais.

## **5.2. Comparação de resultados**

Os valores obtidos para a realização da comparação de resultados foram obtidos a partir de análises feitas a partir do cálculo analítico através de uma folha de Excel (Anexo D), do cálculo através da aplicação desenvolvida em MATLAB e através do programa comercial HAP da Carrier (Anexo E).

Na introdução dos dados referentes ao caso de estudo, no programa HAP só é permitido introduzir dois componentes diferentes nos campos destinados às partições. Devido a esta limitação do programa, foram unicamente introduzidos os valores da Parede interior 1 e 2. Os valores das transferências de calor ocorridas no componente Porta (Figura 5.1-6) foram considerados iguais aos valores obtidos no cálculo analítico e na aplicação em MATLAB. Nos relatórios obtidos no HAP não está discriminado o valor da transferência de calor ocorrida em cada elemento nos campos das partições.

Na comparação dos resultados foram realizados os cálculos das cargas térmicas de arrefecimento e aquecimento nas diferentes análises. Fazendo uma comparação quantitativa entre a aplicação desenvolvida com o cálculo analítico, e da aplicação desenvolvida com o HAP. A Tabela 5.6 contém os valores dos diferentes tipos de cargas térmicas de arrefecimento.

Nos valores obtidos da Tabela 5.6 verifica-se uma discrepância no cálculo das cargas térmicas de arrefecimento referente aos elementos da envolvente exterior. A Figura 5.2 representa o resultado das cargas térmicas de arrefecimento da envolvente exterior obtido pelas três análises efectuadas.

Tabela 5.6. Cargas térmicas de arrefecimento.

Cargas de arrefecimento					
Elemento	Cálculo analítico (W)	Aplicação em MATLAB (W)	HAP (W)	Erro (%)	
				Analítico	HAP
Envolvente exterior					
Parede exterior 1	72,924	72,963	46,000	0,053	58,615
Parede exterior 2	31,105	31,097	28,000	0,026	11,061
Cobertura	112,951	113,025	84,000	0,066	34,554
Pavimento	222,750	222,750	180,000	0,000	23,750
Janela - Parede 1	224,400	222,400	181,000	0,891	22,873
Janela - Parede 2	110,000	110,000	89,000	0,000	23,596
Sub-total	774,130	772,235	608,000	0,245	27,012
Envolvente interior					
Parede interior 1	56,670	56,689	73,000	0,033	0,645
Parede interior 2	15,845	15,840		0,032	
Porta - Parede 1	15,621	15,621	15,621	0,000	0,000
Sub-total	88,136	88,150	88,621	0,016	0,532
Ocupação					
Carga sensível	125,000	125,000	125,000	0,000	0,000
Carga latente	91,667	91,667	92,000	0,000	0,362
Sub-total	216,667	216,667	217,000	0,000	0,154
Iluminação					
Carga sensível	262,500	262,500	262,000	0,000	0,191
Equipamentos Eléctricos					
Carga sensível	375,000	375,000	375,000	0,000	0,000
Cargas diversas					
Carga sensível	0,000	0,000	0,000	-	-
Carga latente	0,000	0,000	0,000	-	-
Carga devido a infiltrações de ar					
Carga sensível	0,113	0,113	0,000	0,265	-
Carga latente	0,113	0,113	0,000	0,000	-
Carga total	0,226	0,226	0,000	0,000	-
Carga Térmica de arrefecimento					
Carga sensível	1624,878	1624,997	1458,621	0,007	11,406
Carga latente	91,780	91,780	92,000	0,000	0,239
Carga total	1716,657	1716,777	1550,621	0,007	10,715

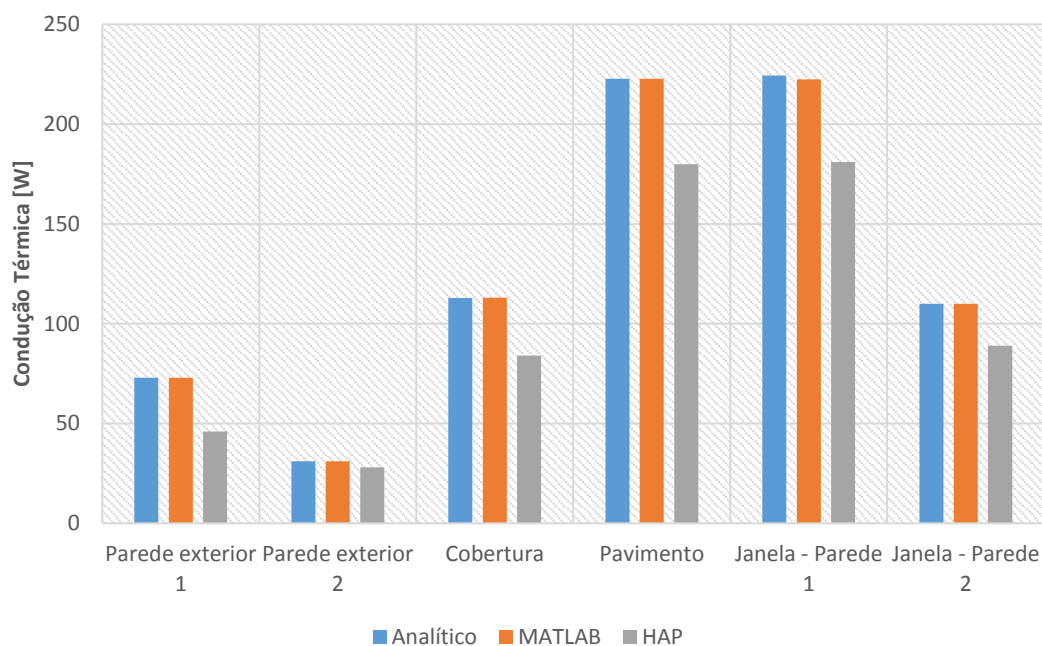


Figura 5.2. Cargas de arrefecimento – Envolvente exterior.

A discrepância verifica-se na comparação entre o Cálculo analítico ou da Aplicação em MATLAB com os valores obtidos a partir do programa HAP, visto o HAP utilizar como metodologia de cálculo o método das funções de transferência. Método esse que revela ser bastante mais assertivo do que a metodologia de cálculo utilizada na aplicação desenvolvida. O cálculo da transferência de calor através da condução térmica da envolvente exterior, através do HAP utiliza um parâmetro denominado de temperatura Sol-air (temperatura que contabiliza o efeito da radiação) e os coeficientes condutivos, na equação da condução da função de transferência [8].

Como seria espectável os valores obtidos da aplicação desenvolvida em MATLAB e os obtidos no cálculo analítico são bastante próximos, devendo-se ao facto de ser utilizado o mesmo método de cálculo nas duas análises. Esta comparação foi realizada com o intuito de fazer a triagem de qualquer erro de programação da função de cálculo utilizada na aplicação desenvolvida.

No cálculo das cargas térmicas internas não são verificadas grandes variações nos resultados obtidos, sendo o erro associado a cada elemento inferior a 1%. Esta aproximação nos resultados é devida à metodologia de cálculo ser igual tanto no cálculo analítico como na aplicação desenvolvida e no HAP. No caso do elemento porta, o valor não foi introduzido no HAP. Embora, a equação utilizada pelo programa para o cálculo

da condução térmica no elemento porta, ser a equação (3.15) no subcapítulo 3.3.2. Comparando o valor obtido pelo HAP relativamente às partições (parede interior 1 e 2) e a soma dos valores obtidos verifica-se um erro inferior a 1%. Na Figura 5.3 estão representadas graficamente as variações existentes no cálculo dos diferentes tipos de cargas térmicas de arrefecimento.

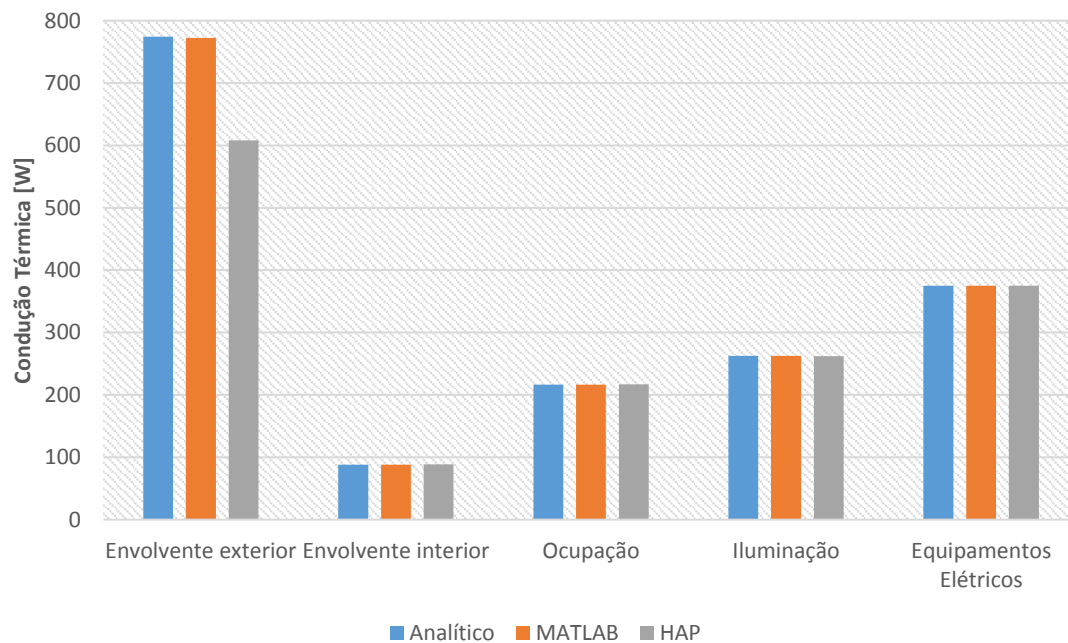


Figura 5.3. Cargas de arrefecimento dos diferentes tipos de cargas térmicas.

No cálculo das cargas térmicas de aquecimento não foram considerados elementos da envoltente interior, visto que as condições estipuladas no caso de estudo para a temperatura do espaço adjacente são constantes ao longo do ano. Com isto, na envoltente interior não existem perdas térmicas, mas sim, ganhos térmicos. Sendo que os ganhos térmicos não são contabilizados nas cargas térmicas de aquecimento. Os valores negativos da carga térmica de aquecimento são meramente indicativos da perda térmica do espaço. A Tabela 5.7 contém os valores dos diferentes tipos de cargas térmicas de aquecimento.

Tabela 5.7. Cargas térmicas de aquecimento.

Cargas de aquecimento					
Elemento	Cálculo analítico (W)	Aplicação em MATLAB (W)	HAP (W)	Erro (%)	
				Analítico	HAP
Envolvente exterior					
Parede exterior 1	-112,700	-112,761	-113,000	0,05	0,21
Parede exterior 2	-48,071	-48,059	-48,000	0,02	0,12
Cobertura	-174,560	-174,675	-175,000	0,07	0,19
Pavimento	-344,250	-344,250	-344,000	0,00	0,07
Janela - Parede 1	-346,800	-346,800	-347,000	0,00	0,06
Janela - Parede 2	-170,000	-170,000	-170,000	0,00	0,00
Sub-total	-1196,381	-1196,545	-1197,000	0,01	0,04
Envolvente interior					
Parede interior 1	0,000	0,000	0,000	0,00	0,00
Parede interior 2	0,000	0,000	0,000	0,00	0,00
Porta - Parede 1	0,000	0,000	0,000	0,00	0,00
Sub-total	0,000	0,000	0,000	0,00	0,00
Carga devido a infiltrações de ar					
Carga sensível	-0,174	-0,174	0,000	0	-
Carga latente	-0,085	-0,085	0,000	0	-
Carga total	-0,259	-0,259	0,000	0	-
Sub-total	-0,518	-0,518	0,000	0	-
Carga Térmica de arrefecimento					
Carga sensível	-1196,556	-1196,719	-1196,000	0,01	0,06
Carga latente	-0,085	-0,085	0,000	0,00	-
Carga total	-1196,641	-1196,804	-1196,000	0,01	0,07

Na comparação dos valores obtidos nas diferentes análises, verifica-se pequenas discrepâncias entre as análises, com os erros inferiores a 1% em cada uma das cargas existentes. Não existem grandes variações visto que a metodologia de cálculo é igual na aplicação em MATLAB, no cálculo analítico e no HAP. A Figura 5.4 representa graficamente as variações existentes no cálculo das cargas térmicas de aquecimento da envolvente exterior nas três análises efectuadas.

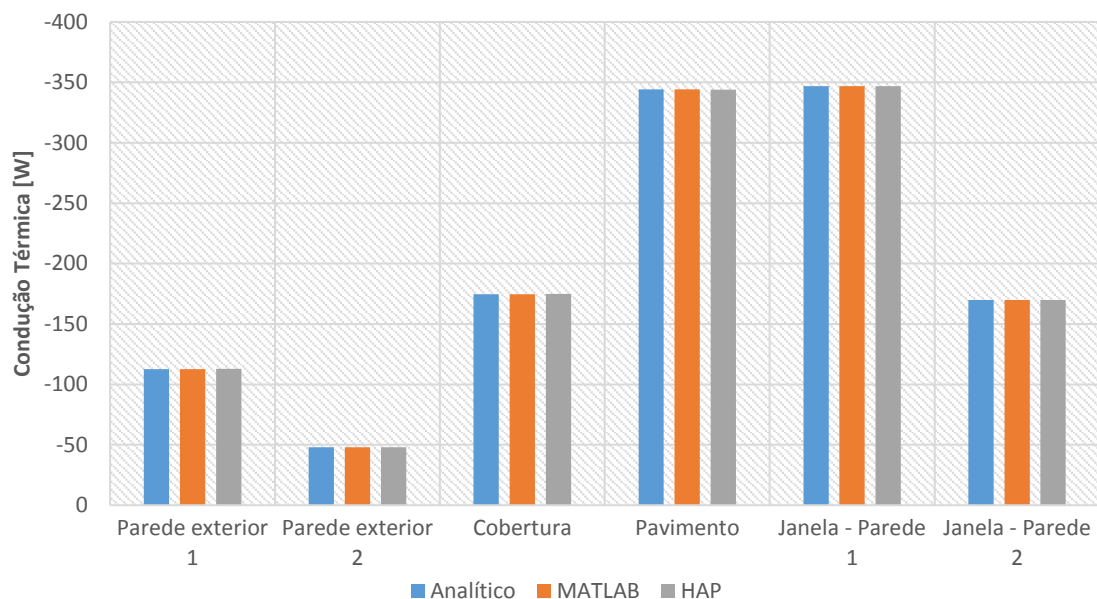


Figura 5.4. Cargas de aquecimento – Envolvente exterior.

Para além do cálculo das cargas térmicas de aquecimento e arrefecimento de projecto, também foi realizado o cálculo de cargas térmicas horárias, isto é, em cada hora do ano. Uma vez que o programa HAP apenas disponibiliza o cálculo das cargas térmicas para um dia modelo do mês, a comparação entre os valores obtidos do HAP é feita com os valores de 1, 15 e 31 de Julho obtidos da aplicação desenvolvida em MATLAB. A Figura 5.5 representa graficamente as variações existentes no cálculo das cargas térmicas horárias.

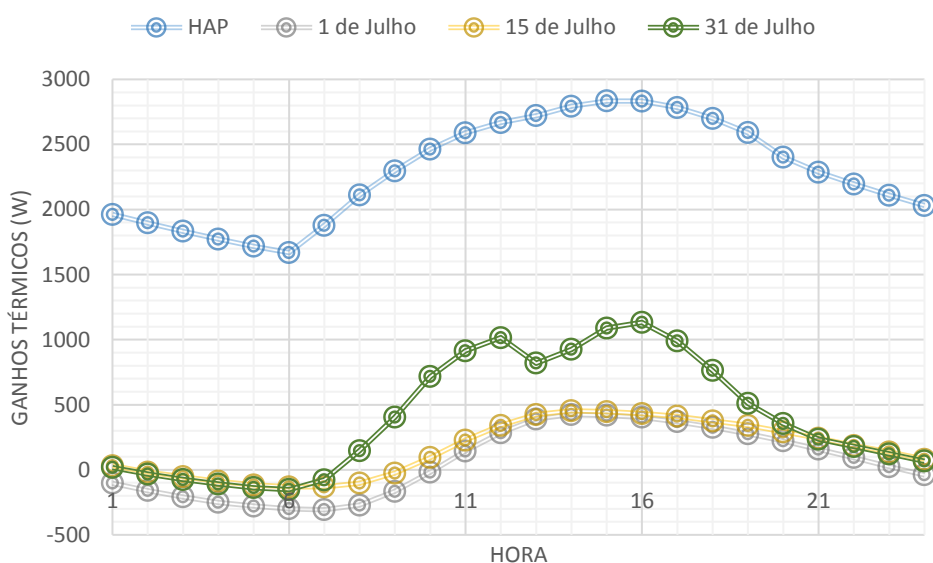


Figura 5.5. Carga Térmica total – Análise horária.



Por análise da representação gráfica do comportamento da condução térmica do espaço, verifica-se que o comportamento térmico é semelhante, embora a magnitude do seu valor seja bastante diferente. Nos valores obtidos pelo programa HAP estão contabilizados os ganhos térmicos de condução, mas também contém a parcela da radiação. Analisando uma hora onde não exista radiação térmica, como por exemplo à 1 hora, verifica-se que o valor dos ganhos térmicos continua superior ao que foi calculado nas cargas térmicas de projecto (valor máximo de condução térmica). Com isto, verifica-se que existem outros factores para além da radiação solar a causar a discrepância de valores. Um dos aspectos fundamentais é o método de cálculo das cargas térmicas utilizado no programa HAP, que emprega o método das funções de transferência (TFM) que é um método de cálculo de cargas térmicas mais rigoroso do que o utilizado na aplicação desenvolvida em MATLAB. O TFM aplica uma série de coeficientes ponderativos. São aplicados aos vários elementos opacos da envolvente exterior os coeficientes condutivos da transferência de calor (CTF). Sendo utilizadas a temperatura Sol-air e a temperatura interior do espaço, para o cálculo do ganho térmico com a devida reflexão da inercia térmica das superfícies dos elementos da envolvente exterior opaca. Depois o TFM aplica uma segunda série de factores ponderativos, os coeficientes de espaço da função de transferência (RTF) ao ganho térmico, e a carga térmica de arrefecimento de todos os elementos, em que é contabilizada a radiação térmica (envolvente exterior). Com o objectivo de determinar o efeito do armazenamento térmico na conversão do ganho térmico em carga térmica de arrefecimento [8]. Outro factor é o ganho de calor a uma determinada hora ser convertida em carga térmica na hora actual e num número de horas seguintes. A energia absorvida pelas paredes, chão, entre outros componentes da envolvente de um espaço, contribui para a carga térmica apenas após um hiato de tempo. Parte desta energia contínua presente e a irradiar novamente calor para o espaço, mesmo quando as fontes de calor tenham sido desligadas ou removidas do espaço [8].

O cálculo das cargas térmicas horárias, da aplicação desenvolvida em MATLAB, foi realizado através da mesma metodologia de cálculo utilizado no das cargas térmicas de arrefecimento ou aquecimento de projecto. A selecção do tipo de carga é realizada, consoante a temperatura exterior seja menor ou maior que a temperatura interior do espaço. O conceito introduzido no cálculo das cargas internas é o conceito dos padrões de referência de utilização dos edifícios, logo os únicos valores que variam ao longo do tempo são a temperatura de bolbo seco exterior e a humidade relativa exterior.

Na aplicação desenvolvida é unicamente calculada a carga térmica condutiva consoante as condições exteriores e os padrões de referência de utilização dos edifícios para aquele instante de uma hora.

## 6. Conclusões e desenvolvimentos futuros

No decurso desta dissertação foi desenvolvida uma aplicação informática, que efectua o cálculo da condução térmica em edifícios. Esta aplicação integra as estruturas de dados com informação sobre tipologias de construção nacionais, interface gráfica e funções com o intuito de tornar a aplicação acessível e apelativa a nível estético aos utilizadores.

A aplicação desenvolvida permite: a criação, exportação, importação e eliminação de múltiplos projectos. Possibilita a introdução de: novos locais geográficos, materiais de construção de edifícios, elementos construtivos, espaços arquitectónicos, e horários de padrões de referência de utilização de edifícios. Tendo como informação pré-definida uma estrutura de dados contendo 139 materiais de construção distintos, agrupados em 17 classes. A estrutura de dados pré-definida também é composta por informação sobre: vidros simples, duplos, triplos caixilhos, janelas simples, janelas duplas, paredes de fachada, pavimentos e coberturas. Relativamente aos horários de ocupação, iluminação e de equipamentos eléctricos foram definidos para as seguintes tipologias: Hipermercado, Venda por grosso, Centro comercial, Pequenas lojas, Restaurantes, Pastelarias, Cinemas e teatros, Escritórios, Bibliotecas, Museus e galerias, Tribunais, Ministérios e Câmaras, Estabelecimentos de ensino, Estabelecimento de ensino superior, Estabelecimento de saúde sem internamento, Estabelecimento de saúde com internamento. Os locais geográficos acessíveis pela estrutura de dados pré-definida foram as capitais de distrito (Portugal continental), Ponta Delgada e Funchal. Esta aplicação efectua o cálculo da condução térmica de aquecimento e arrefecimento para condições de projecto e para análise horária.

O cálculo da condução térmica é realizado através de um modelo matemático, descrito detalhadamente no capítulo 3. Pelo estudo dos métodos utilizados no cálculo das cargas térmicas em programas verifica-se que, quanto maior for a precisão do método, maior é o grau de dificuldade em implementá-lo.

Nas análises elaboradas através do cálculo analítico, cálculo através da aplicação desenvolvida em MATLAB e com recurso ao programa HAP verifica-se que nem todos os valores obtidos estão em conformidade. No que diz respeito às cargas térmicas de arrefecimento, no cálculo de projecto existem discrepâncias nos valores dos elementos da envolvente externa. Esta variação entre os valores obtidos através do cálculo analítico ou da aplicação desenvolvida em MATLAB, com os valores obtidos do programa HAP

devem-se à sua diferente metodologia. Nas duas primeiras análises (cálculo analítico e através da aplicação desenvolvida) foi considerado no seu cálculo a temperatura exterior e no programa HAP o conceito de temperatura sol-air e os coeficientes condutivos e de espaço da função de transferência. Os erros associados aos elementos da envolvente externa variam entre 11% a 59%. Nos elementos considerados vãos envidraçados (da envolvente externa) o único factor são os coeficientes de espaço da função de transferência. Para os vãos envidraçados obtiveram-se erros na ordem dos 20%. Nos restantes tipos de cargas térmicas os valores obtidos são bastante próximos, tendo-se obtido erros associados às comparações com valores inferiores a 1%.

No cálculo das cargas térmicas de aquecimento de projecto, os valores são bastante próximos, tendo sido obtidos erros inferiores a 1% nas comparações das análises. Esta proximidade de valores deve-se substancialmente ao facto de nas três análises a metodologia de cálculo ser exactamente igual.

Quanto ao cálculo das cargas térmicas horárias verifica-se uma grande discrepância entre os valores obtidos através da aplicação desenvolvida em MATLAB e os do programa HAP. Esta variação de valores deve-se ao facto de no HAP serem contabilizados os ganhos térmicos devido à radiação solar, enquanto a metodologia de cálculo (TFM) do programa HAP é mais rigorosa do que a metodologia utilizada na aplicação desenvolvida em MATLAB. Também se deve ao facto de o ganho de calor de uma determinada hora ser convertido em carga térmica na hora actual e num número de horas seguintes. Ou seja, a energia absorvida pelas paredes, chão, entre outros componentes da envolvente de um espaço, contribui para a carga térmica apenas depois de um hiato de tempo. Parte desta energia continua presente e a irradiar novamente calor para o espaço, mesmo quando as fontes de calor tenham sido deligadas ou removidas do espaço [8].

Em relação ao cálculo de cargas térmicas condutivas para o projecto, pela análise comparativa dos resultados, a aplicação não é invalidada. Visto que nas cargas térmicas de arrefecimento a aplicação é menos precisa por excesso, no entanto não contribuirá para uma escolha diferente do equipamento seleccionado. No cálculo das cargas térmicas de aquecimento a aplicação é válida, visto os erros obtidos entre os valores da aplicação e os do programa de referência serem inferiores a 1 %. Na análise horária das cargas térmicas de um espaço a aplicação não pode ser considerada válida, uma vez que apresenta valores bastante dispares entre a aplicação e o HAP.

Como desenvolvimentos futuros são apresentados os seguintes pontos:

- Melhorar a qualidade das imagens do ISEL e da ADEM no GUI inicial;
- Permitir a modificação do “Nome de projecto”, “Nome da instituição” e o “Nome do autor”;
- Criar algumas mensagens de erros e definir valores às variáveis quando não são preenchidas as caixas de texto quando seleccionada a opção guardar informação;
- Completar a informação da estrutura de dados pré-definida com informação relativa a ganhos térmicos de ocupação, equipamentos eléctricos, iluminação e cargas diversas;
- Melhorar o aspecto visual das janelas “Definir Horário” e “Relatórios”;
- Melhorar a metodologia de cálculo das cargas térmicas de arrefecimento da envolvente exterior;
- Adoptar um método de maior precisão como por exemplo o método das funções de transferência ou até mesmo o método das séries radiactivas. Uma vez que são métodos que aproximam com precisão os resultados à realidade do espaço em análise;
- Melhorar a estética do ficheiro que recebe os dados do projecto exportado.



## Referências Bibliográficas

- [1] J. . Holman, *Heat transfer*, 10<sup>a</sup> Edição. McGraw-Hill, 2010.
- [2] V. Monteiro, *Refrigeração I - Técnicas e Competências Ambientais: Bases e Fundamentos*, 1<sup>a</sup> Edição. ETEP - Edições Técnicas e Profissionais, 2015.
- [3] C. A. Pina dos Santos e L. Matias, *Coeficientes de transmissão térmica de elementos da envolvente dos edifícios*, 7<sup>a</sup> Edição. Lisboa, 2007.
- [4] T. Bergman, A. Lavine, F. Incropera, e D. Dewitt, *Fundamentals of Heat and Mass Transfer*, 7<sup>a</sup> Edição. John Wiley & Sons, 2011.
- [5] L. Roriz, *Climatização - Concepção, Instalação e Condução de Sistemas*, 2<sup>a</sup> Edição. Orion, 2007.
- [6] A. P. de A. M. Alves, “Simulação térmica multizona de um edifício escolar : Simulação térmica multizona de um edifício escolar : uma comparação de programas,” Universidade do Minho, 2011.
- [7] J. Aswegan, J. Connell, P. Simmonds, e J. Traylor, *ASHRAE Fundamentals Handbook 2013*. ASHRAE, Inc., 2013.
- [8] L. Berglund, K. Brian, R. Pons, e G. Reeves, *ASHRAE Fundamental Handbook 1997*. ASHRAE, Inc., 1997.
- [9] S. K. Wang, *Handbook of Air Conditioning and Refrigeration*, 2<sup>a</sup> Edição. McGraw-Hill, 2000.
- [10] Carrier, “Hourly Analysis Program,” 2016. [Online]. Disponível: <https://www.carrier.com/commercial/en/us/software/hvac-system-design/hourly-analysis-program/>. [Acedido: 18-Jul-2016].
- [11] EnergyPlus, “EnergyPlus,” 2016. [Online]. Disponível: <https://energyplus.net/extras>. [Acedido: 18-Jul-2016].
- [12] JH and LBNL, “DOE2,” 2016. [Online]. Available: <http://www.doe2.com/>. [Acedido: 18-Jul-2016].
- [13] TRNSYS, “Transient System Simulation Tool,” 2016. [Online]. Disponível: <http://www.trnsys.com/>. [Acedido: 18-Jul-2016].
- [14] TRANE, “Trace 700 - HVAC load design and analysis software,” 2015.

- [15] TRANE, “Trace 700,” 2016. [Online]. Disponível: <http://www.trane.com/commercial/north-america/us/en/products-systems/design-and-analysis-tools/analysis-tools/trace-700/trace-700-program-features.html#1>. [Acedido: 18-Jul-2016].
- [16] Universidade do Minho, “Capítulo XVIII revestimento de pavimentos,” 2016. [Online]. Disponível: [http://www.civil.uminho.pt/lftc/Textos\\_files/construcoes/cp2/Cap. XVIII - Revestimentos de Pavimentos..pdf](http://www.civil.uminho.pt/lftc/Textos_files/construcoes/cp2/Cap. XVIII - Revestimentos de Pavimentos..pdf). [Acedido: 18-Jul-2016].
- [17] I. Flores-Colen, L. G. Silva, J. Garcia, A. Silva, and N. Neto, “Revestimentos de pisos.” [Online]. Disponível: <http://www.civil.ist.utl.pt/~joaof/tc-pb/22 Revestimentos de pisos - 25%C2%AA e 26%C2%AA aulas te%C3%B3ricas.pdf>. [Acedido: 18-Jul-2016].
- [18] Instituto Politécnico de Tomar, “Coberturas de Edifícios.” [Online]. Disponível: [http://www.estt.ipt.pt/download/disciplina/1136\\_\\_CoberturasPlanas.pdf](http://www.estt.ipt.pt/download/disciplina/1136__CoberturasPlanas.pdf). [Acedido: 18-Jul-2016].
- [19] J. M. F. da S. Rocha, “Reabilitação do ponto de vista térmico de coberturas inclinadas, no Centro Histórico do Porto,” Faculdade de Engenharia da Universidade do Porto, 2008.
- [20] V. G. P. Mendes, “Quantificação do coeficiente de transmissão térmica de vãos envidraçados - Modelo de cálculo,” Faculdade de Engenharia de Universidade do Porto, 2011.
- [21] BS EN ISO, “BS EN ISO 10077-1:2006 Thermal performance of windows , doors and shutters - Calculation of thermal transmittance,” 2006.
- [22] BS EN ISO, “BS EN ISO 10077-2:2012 Thermal performance of windows , doors and shutters - Calculation of thermal transmittance Part 2 : Numerical method for frames,” 2012.
- [23] J. Frade, “Climatização geral,” 2006.
- [24] “Air Pressure at Altitude Calculator.” [Online]. Disponível: <http://www.mide.com/pages/air-pressure-at-altitude-calculator>. [Acedido: 18-Jul-2016].
- [25] Ministério das Obras Públicas, “O Regulamento dos Sistemas Energéticos de Climatização em Edifícios (RSECE)-Decreto-Lei n.º 79/2006,” *Diário da República*, no. 4 de Abril. p. 53 (2416-2468), 2006.



# Anexos

(Todos os anexos estão disponíveis na versão digital)

**Anexo A – Estrutura de dados**

**Anexo B – Programação em ambiente gráfico**

**Anexo C – Funções**

**Anexo D – Cargas térmicas de aquecimento e arrefecimento  
(Cálculo Analítico)**

**Anexo E – Relatórios do programa HAP**

**Anexo F – Comparação entre os valores do HAP e da aplicação  
nas cargas térmicas horárias**

**Anexo G – Relatório Análise de Projecto**

**Anexo H – Relatório Análise Anual**



## **Anexo A – Bases de dados**



```
%Base de dados de materiais
```

```
% Dados relativos a Água
```

```
global material
```

```
%Água a 10°C
```

```
material.agua(1).nome = 'Água a 10°C';
```

```
material.agua(1).classificacao = 'Água';
```

```
material.agua(1).rho = 1000;
```

```
%kg/m^3 retirado:ITE 50
```

```
material.agua(1).k = 0.60;
```

```
%W/(m*°C) retirado:ITE 50
```

```
%Água a 40°C
```

```
material.agua(2).nome = 'Água a 40°C';
```

```
material.agua(2).classificacao = 'Água';
```

```
material.agua(2).rho = 990;
```

```
%kg/m^3 retirado:ITE 50
```

```
material.agua(2).k = 0.63;
```

```
%W/(m*°C) retirado:ITE 50
```

```
%Base de dados de materiais
```

```
% Dados relativos a Argamassas (cimento ou cal)de reboco e de assentamento de✓  
tijolos e de blocos
```

```
global material
```

```
%Argamassas e rebocos tradicionais
```

```
material.argamassa(1).nome = 'Argamassas e rebocos tradicionais';  
material.argamassa(1).classificacao = 'Argamassas (cimento ou cal)de✓  
reboco e de assentamento de tijolos e de blocos';  
material.argamassa(1).rho = 1900; %kg/m^3 retirado:✓  
ITE 50  
material.argamassa(1).k = 1.3; %W/(m*°C) retirado:✓  
ITE 50
```

```
%Argamassas e rebocos não-tradicionais
```

```
material.argamassa(2).nome = 'Argamassas e rebocos não-tradicionais';  
material.argamassa(2).classificacao = 'Argamassas (cimento ou cal)de✓  
reboco e de assentamento de tijolos e de blocos';  
material.argamassa(2).rho = 625; %kg/m^3 retirado:✓  
ITE 50  
material.argamassa(2).k = 0.30; %W/(m*°C) retirado:✓  
ITE 50
```

```
%Argamassas e rebocos de cal e areia ou de argamassa bastarda
```

```
material.argamassa(3).nome = 'Argamassas e rebocos de cal e areia ou de✓  
argamassa bastarda';  
material.argamassa(3).classificacao = 'Argamassas (cimento ou cal)de✓  
reboco e de assentamento de tijolos e de blocos';  
material.argamassa(3).rho = 1600; %kg/m^3 retirado:✓  
ITE 50  
material.argamassa(3).k = 0.80; %W/(m*°C) retirado:✓  
ITE 50
```

```

%Base de dados de materiais

% Dados relativos a Betões

global material

%Betão de inertes correntes (calcários, siliciosos e silico-calcários)

%Betão normal

material.betao(1).nome = 'Betão normal';
material.betao(1).classificacao = 'Betão de inertes correntes✓
(calcários, siliciosos e silico-calcários)';
material.betao(1).rho = 2150; %kg/m^3 retirado:ITE 50
material.betao(1).k = 1.65; %W/(m*°C) retirado:ITE 50

%Betão cavernoso

material.betao(2).nome = 'Betão cavernoso';
material.betao(2).classificacao = 'Betão de inertes correntes✓
(calcários, siliciosos e silico-calcários)';
material.betao(2).rho = 1700; %kg/m^3 retirado:ITE 50
material.betao(2).k = 1.15; %W/(m*°C) retirado:ITE 50

%Betão armado de inertes correntes (calcários, siliciosos e silico-calcários)

%Betão com percentagem de armadura <1% (em volume)

material.betao(3).nome = 'Betão com percentagem de armadura <1% (em✓
volume)';
material.betao(3).classificacao = 'Betão armado de inertes correntes✓
(calcários, siliciosos e silico-calcários)';
material.betao(3).rho = 2350; %kg/m^3 retirado:ITE 50
material.betao(3).k = 2.0; %W/(m*°C) retirado:ITE 50

%Betão com percentagem significativa de armadura paralela ao fluxo de✓
calor

%Betão com percentagem de armadura: 1-2% (em volume)

material.betao(4).nome = 'Betão com percentagem de armadura paralela ao✓
fluxo de calor: 1-2% (em volume)';
material.betao(4).classificacao = 'Betão armado de inertes correntes✓
(calcários, siliciosos e silico-calcários)';
material.betao(4).rho = 2350; %kg/m^3 retirado:ITE 50
material.betao(4).k = 2.3; %W/(m*°C) retirado:ITE 50

%Betão com percentagem de armadura: >2% (em volume)

material.betao(5).nome = 'Betão com percentagem de armadura paralela ao✓
fluxo de calor: >2% (em volume)';
material.betao(5).classificacao = 'Betão armado de inertes correntes✓
(calcários, siliciosos e silico-calcários)';
material.betao(5).rho = 2500; %kg/m^3 retirado:ITE 50
material.betao(5).k = 2.5; %W/(m*°C) retirado:ITE 50

%Betão de inertes de argila expandida - Betão estrutural

```

```

%Betão estrutural, com areia leve e sem areia do rio

material.betao(6).nome = 'Betão estrutural, com areia leve e sem areia
do rio';
material.betao(6).classificacao = 'Betão de inertes de argila
expandida';
material.betao(6).rho = 1500;           %kg/m^3   retirado:ITE 50
material.betao(6).k = 0.85;             %W/(m*°C) retirado:ITE 50

%Betão estrutural, sem areia leve e com areia do rio

material.betao(7).nome = 'Betão estrutural, sem areia leve e com areia
do rio';
material.betao(7).classificacao = 'Betão de inertes de argila
expandida';
material.betao(7).rho = 1700;           %kg/m^3   retirado:ITE 50
material.betao(7).k = 1.05;             %W/(m*°C) retirado:ITE 50

%Betão isolante "resistente"

%Betão isolante "resistente", com areia leve e sem areia do rio

material.betao(8).nome = 'Betão isolante resistente, com areia leve e
sem areia do rio';
material.betao(8).classificacao = 'Betão isolante resistente';
material.betao(8).rho = 1000;           %kg/m^3   retirado:ITE 50
material.betao(8).k = 0.36;             %W/(m*°C) retirado:ITE 50

%Betão isolante "resistente", com areia leve e areia do rio (menor ou
igual a 10%)

material.betao(9).nome = 'Betão isolante "resistente", com areia leve e
areia do rio (menor ou igual a 10%)';
material.betao(9).classificacao = 'Betão isolante resistente';
material.betao(9).rho = 1300;           %kg/m^3   retirado:ITE 50
material.betao(9).k = 0.70;             %W/(m*°C) retirado:ITE 50

%Betão cavernoso ou semi-cavernoso

%Betão cavernoso ou semi-cavernoso, com areia leve e sem areia do rio

material.betao(10).nome = 'Betão cavernoso ou semi-cavernoso, com areia
leve e sem areia do rio';
material.betao(10).classificacao = 'Betão cavernoso ou semi-cavernoso';
material.betao(10).rho = 900;           %kg/m^3   retirado:ITE 50
material.betao(10).k = 0.33;            %W/(m*°C) retirado:ITE 50

%Betão cavernoso ou semi-cavernoso, sem areia (leve ou do rio) e com
fraca dosagem de cimento

material.betao(11).nome = 'Betão cavernoso ou semi-cavernoso, sem areia
(leve ou do rio) e com fraca dosagem de cimento';
material.betao(11).classificacao = 'Betão cavernoso ou semi-cavernoso';
material.betao(11).rho = 400;           %kg/m^3   retirado:ITE 50
material.betao(11).k = 0.14;            %W/(m*°C) retirado:ITE 50

%Betão de inertes de perlite ou de vermiculite expandida (3 a 6 mm) fabricado

```



em obra

%dosagem cimento/inertes 1/6

```
material.betao(12).nome = 'Betão de inertes de perlite ou de vermiculite✓  
expandida - dosagem cimento/inertes 1/6';  
material.betao(12).classificacao = 'Betão de inertes de perlite ou de✓  
vermiculite expandida (3 a 6 mm) fabricado em obra';  
material.betao(12).rho = 500; %kg/m^3 retirado:ITE 50  
material.betao(12).k = 0.24; %W/(m*°C) retirado:ITE 50
```

%dosagem cimento/inertes 1/3

```
material.betao(13).nome = 'Betão de inertes de perlite ou de vermiculite✓  
expandida - dosagem cimento/inertes 1/3';  
material.betao(13).classificacao = 'Betão de inertes de perlite ou de✓  
vermiculite expandida (3 a 6 mm) fabricado em obra';  
material.betao(13).rho = 700; %kg/m^3 retirado:ITE 50  
material.betao(13).k = 0.31; %W/(m*°C) retirado:ITE 50
```

%Betão de inertes de pedra-pomes

```
material.betao(14).nome = 'Betão de Inertes de pedra-pomes';  
material.betao(14).classificacao = 'Betão de Inertes de pedra-pomes';  
material.betao(14).rho = 1050; %kg/m^3 retirado:ITE 50  
material.betao(14).k = 0.46; %W/(m*°C) retirado:ITE 50
```

%Betão de inertes de pedra-pomes para blocos de alvenaria

```
material.betao(15).nome = 'Betão de inertes de pedra-pomes para blocos✓  
de alvenaria';  
material.betao(15).classificacao = 'Betão de inertes de pedra-pomes para✓  
blocos de alvenaria';  
material.betao(15).rho = 500; %kg/m^3 retirado:ITE 50  
material.betao(15).k = 0.16; %W/(m*°C) retirado:ITE 50
```

%Betão de inertes de poliestireno expandido

```
material.betao(16).nome = 'Betão de inertes de poliestireno expandido';  
material.betao(16).classificacao = 'Betão de inertes de poliestireno✓  
expandido';  
material.betao(16).rho = 500; %kg/m^3 retirado:ITE 50  
material.betao(16).k = 0.18; %W/(m*°C) retirado:ITE 50
```

%Betão celular autoclavado

```
material.betao(17).nome = 'Betão celular autoclavado';  
material.betao(17).classificacao = 'Betão celular autoclavado';  
material.betao(17).rho = 450; %kg/m^3 retirado:ITE 50  
material.betao(17).k = 0.16; %W/(m*°C) retirado:ITE 50
```

%Betão de terra estabilizada

```
material.betao(18).nome = 'Betão de terra estabilizada';  
material.betao(18).classificacao = 'Betão de terra estabilizada';  
material.betao(18).rho = 1985; %kg/m^3 retirado:ITE 50  
material.betao(18).k = 1.1; %W/(m*°C) retirado:ITE 50
```

```
%Base de dados de materiais
```

```
% Dados relativos a Borrachas
```

```
global material
```

```
%Borracha natural
```

```
material.borrachas(1).nome = 'Borracha natural';  
material.borrachas(1).classificacao = 'Borrachas';  
material.borrachas(1).rho = 910; %kg/m^3 retirado:ITE✓
```

```
50
```

```
material.borrachas(1).k = 0.13; %W/(m*°C) retirado:ITE✓
```

```
50
```

```
%Neopreno (policloropreno)
```

```
material.borrachas(2).nome = 'Neopreno (policloropreno)';  
material.borrachas(2).classificacao = 'Borrachas';  
material.borrachas(2).rho = 1240; %kg/m^3 retirado:ITE✓
```

```
50
```

```
material.borrachas(2).k = 0.23; %W/(m*°C) retirado:ITE✓
```

```
50
```

```
%Borracha butílica (isobuteno)
```

```
material.borrachas(3).nome = 'Borracha butílica (isobuteno)';  
material.borrachas(3).classificacao = 'Borrachas';  
material.borrachas(3).rho = 1200; %kg/m^3 retirado:ITE✓
```

```
50
```

```
material.borrachas(3).k = 0.24; %W/(m*°C) retirado:ITE✓
```

```
50
```

```
%Monómero de etileno propileno dieno (EPDM)
```

```
material.borrachas(4).nome = 'Monómero de etileno propileno dieno  
(EPDM)';  
material.borrachas(4).classificacao = 'Borrachas';  
material.borrachas(4).rho = 1150; %kg/m^3 retirado:ITE✓
```

```
50
```

```
material.borrachas(4).k = 0.25; %W/(m*°C) retirado:ITE✓
```

```
50
```

```
%Poli-isobutileno
```

```
material.borrachas(5).nome = 'Poli-isobutileno';  
material.borrachas(5).classificacao = 'Borrachas';  
material.borrachas(5).rho = 930; %kg/m^3 retirado:ITE✓
```

```
50
```

```
material.borrachas(5).k = 0.20; %W/(m*°C) retirado:ITE✓
```

```
50
```

```
%Polisulfureto
```

```
material.borrachas(6).nome = 'Polisulfureto';  
material.borrachas(6).classificacao = 'Borrachas';  
material.borrachas(6).rho = 1700; %kg/m^3 retirado:ITE✓
```

```
50
```

```
material.borrachas(6).k = 0.40; %W/(m*°C) retirado:ITE✓
```

```
50
```

%Butadieno

material.borrachas(7).nome = 'Polisulfureto';

material.borrachas(7).classificacao = 'Borrachas';

material.borrachas(7).rho = 980; %kg/m<sup>3</sup> retirado:ITE✓

50

material.borrachas(7).k = 0.25; %W/(m\*°C) retirado:ITE✓

50

```
%Base de dados de materiais
```

```
% Dados relativos a Fibrocimento
```

```
global material
```

```
%Placas de fibrocimento com fibras de amianto
```

```
material.fibrocimento(1).nome = 'Placas de fibrocimento com fibras de amianto';  
material.fibrocimento(1).classificacao = 'Fibrocimento';  
material.fibrocimento(1).rho = 1600; %kg/m^3 retirado:ITE  
50  
material.fibrocimento(1).k = 0.65; %W/(m*°C) retirado:ITE  
50
```

```
%Placas de fibrocimento com fibras celulósicas
```

```
material.fibrocimento(2).nome = 'Placas de fibrocimento com fibras celulósicas';  
material.fibrocimento(2).classificacao = 'Fibrocimento';  
material.fibrocimento(2).rho = 1200; %kg/m^3 retirado:ITE  
50  
material.fibrocimento(2).k = 0.35; %W/(m*°C) retirado:ITE  
50
```

```
%Base de dados de materiais
```

```
% Dados relativos a Gases
```

```
global material
```

```
%Ar
```

```
material.gases(1).nome = 'Ar';  
material.gases(1).classificacao = 'Gases';  
material.gases(1).rho = 1.23;
```

```
%kg/m^3 retirado:ITE✓
```

```
material.gases(1).k = 0.025;
```

```
%W/(m*°C) retirado:ITE✓
```

```
%Anidrido carbónico
```

```
material.gases(2).nome = 'Anidrido carbónico';  
material.gases(2).classificacao = 'Gases';  
material.gases(2).rho = 1.95;
```

```
%kg/m^3 retirado:ITE✓
```

```
material.gases(2).k = 0.014;
```

```
%W/(m*°C) retirado:ITE✓
```

```
%Árgon
```

```
material.gases(3).nome = 'Ar';  
material.gases(3).classificacao = 'Gases';  
material.gases(3).rho = 1.70;
```

```
%kg/m^3 retirado:ITE✓
```

```
material.gases(3).k = 0.017;
```

```
%W/(m*°C) retirado:ITE✓
```

```
%Cripton
```

```
material.gases(4).nome = 'Cripton';  
material.gases(4).classificacao = 'Gases';  
material.gases(4).rho = 3.56;
```

```
%kg/m^3 retirado:ITE✓
```

```
material.gases(4).k = 0.0090;
```

```
%W/(m*°C) retirado:ITE✓
```

```
%Xénon
```

```
material.gases(5).nome = 'Xénon';  
material.gases(5).classificacao = 'Gases';  
material.gases(5).rho = 5.68;
```

```
%kg/m^3 retirado:ITE✓
```

```
material.gases(5).k = 0.0054;
```

```
%W/(m*°C) retirado:ITE✓
```

```

%Base de dados de materiais

% Dados relativos a Gessos (Estuques)

global material

%Estuques sem inertes

    %Estuque tradicional

    material.gessos(1).nome = 'Estuque tradicional';
    material.gessos(1).classificacao = 'Estuques sem inertes';
    material.gessos(1).rho = 900; %kg/m^3 retirado:ITE 50
    material.gessos(1).k = 0.40; %W/(m*°C) retirado:ITE 50

    %Estuque projectado, estuque fino, estuque de elevada dureza

    material.gessos(2).nome = 'Estuque projectado, estuque fino, estuque de
elevada dureza';
    material.gessos(2).classificacao = 'Estuques sem inertes';
    material.gessos(2).rho = 500; %kg/m^3 retirado:ITE 50
    material.gessos(2).k = 0.18; %W/(m*°C) retirado:ITE 50

    %Estuque de gesso e areia

    material.gessos(3).nome = 'Estuque de gesso e areia';
    material.gessos(3).classificacao = 'Estuque de gesso e areia';
    material.gessos(3).rho = 1500; %kg/m^3 retirado:ITE 50
    material.gessos(3).k = 0.80; %W/(m*°C) retirado:ITE 50

    %Estuques com inertes leves e/ou fibras minerais

    %Estuque com grânulos de perlite ou fibras minerais

    material.gessos(4).nome = 'Estuque com grânulos de perlite ou fibras
minerais';
    material.gessos(4).classificacao = 'Estuques com inertes leves e/ou
fibras minerais';
    material.gessos(4).rho = 550; %kg/m^3 retirado:ITE 50
    material.gessos(4).k = 0.18; %W/(m*°C) retirado:ITE 50

    %Estuques com inertes leves e/ou fibras minerais expandidas(1 a 2mm)

    material.gessos(5).nome = 'Estuques com inertes leves e/ou fibras
minerais expandidas(1 a 2mm)';
    material.gessos(5).classificacao = 'Estuques com inertes leves e/ou
fibras minerais';
    material.gessos(5).rho = 750; %kg/m^3 retirado:ITE 50
    material.gessos(5).k = 0.30; %W/(m*°C) retirado:ITE 50

    %Placas de gesso cartonado

    material.gessos(6).nome = 'Placas de gesso cartonado';
    material.gessos(6).classificacao = 'Placas de gesso cartonado';
    material.gessos(6).rho = 875; %kg/m^3 retirado:ITE 50
    material.gessos(6).k = 0.25; %W/(m*°C) retirado:ITE 50

```

```
%Base de dados de materiais
```

```
% Dados relativos a Materiais de impermeabilização e mastiques
```

```
global material
```

```
%Asfalto puro
```

```
material.impermeabilizacao_mastiques(1).nome = 'Asfalto puro';  
material.impermeabilizacao_mastiques(1).classificacao = 'Materiais de  
impermeabilização e mastiques';  
material.impermeabilizacao_mastiques(1).rho = 2000; %kg/m^3✓  
retirado:ITE 50  
material.impermeabilizacao_mastiques(1).k = 0.70; %W/(m*°C)✓  
retirado:ITE 50
```

```
%Asfalto areado
```

```
material.impermeabilizacao_mastiques(2).nome = 'Asfalto areado';  
material.impermeabilizacao_mastiques(2).classificacao = 'Materiais de  
impermeabilização e mastiques';  
material.impermeabilizacao_mastiques(2).rho = 2000; %kg/m^3✓  
retirado:ITE 50  
material.impermeabilizacao_mastiques(2).k = 1.15; %W/(m*°C)✓  
retirado:ITE 50
```

```
%Betume puro
```

```
material.impermeabilizacao_mastiques(3).nome = 'Betume puro';  
material.impermeabilizacao_mastiques(3).classificacao = 'Materiais de  
impermeabilização e mastiques';  
material.impermeabilizacao_mastiques(3).rho = 1000; %kg/m^3✓  
retirado:ITE 50  
material.impermeabilizacao_mastiques(3).k = 0.17; %W/(m*°C)✓  
retirado:ITE 50
```

```
%Membranas flexíveis impregnadas com betume
```

```
material.impermeabilizacao_mastiques(4).nome = 'Membranas flexíveis  
impregnadas com betume';  
material.impermeabilizacao_mastiques(4).classificacao = 'Materiais de  
impermeabilização e mastiques';  
material.impermeabilizacao_mastiques(4).rho = 1050; %kg/m^3✓  
retirado:ITE 50  
material.impermeabilizacao_mastiques(4).k = 0.23; %W/(m*°C)✓  
retirado:ITE 50
```

```
%PVC flexível (40% de plastificante)
```

```
material.impermeabilizacao_mastiques(5).nome = 'PVC flexível (40% de  
plastificante)';  
material.impermeabilizacao_mastiques(5).classificacao = 'Materiais de  
impermeabilização e mastiques';  
material.impermeabilizacao_mastiques(5).rho = 1200; %kg/m^3✓  
retirado:ITE 50  
material.impermeabilizacao_mastiques(5).k = 0.14; %W/(m*°C)✓  
retirado:ITE 50
```

```
%Mastique de silicone puro
```

```

material.impermeabilizacao_mastiques(6).nome = 'Mastique de silicone✓
puro';
material.impermeabilizacao_mastiques(6).classificacao = 'Materiais de✓
impermeabilização e mastiques';
material.impermeabilizacao_mastiques(6).rho = 1200; %kg/m^3✓
retirado:ITE 50
material.impermeabilizacao_mastiques(6).k = 0.35; %W/(m*°C)✓
retirado:ITE 50

%Mastique de silicone

material.impermeabilizacao_mastiques(7).nome = 'Mastique de silicone';
material.impermeabilizacao_mastiques(7).classificacao = 'Materiais de✓
impermeabilização e mastiques';
material.impermeabilizacao_mastiques(7).rho = 1450; %kg/m^3✓
retirado:ITE 50
material.impermeabilizacao_mastiques(7).k = 0.50; %W/(m*°C)✓
retirado:ITE 50

```



```

%Base de dados de materiais

% Dados relativos a inertes, solos e terras

global material

    %Areia, gravilha, seixo, brita

        material.inertes_solos_terras(1).nome = 'Areia, gravilha, seixo, brita';
        material.inertes_solos_terras(1).classificacao = 'Inertes, solos e
terras';
        material.inertes_solos_terras(1).rho = 1950; %kg/m^3
retirado:ITE 50
        material.inertes_solos_terras(1).k = 2.00; %W/(m*°C)
retirado:ITE 50

    %Argila ou lodo

        material.inertes_solos_terras(2).nome = 'Argila ou lodo';
        material.inertes_solos_terras(2).classificacao = 'Inertes, solos e
terras';
        material.inertes_solos_terras(2).rho = 1500; %kg/m^3
retirado:ITE 50
        material.inertes_solos_terras(2).k = 1.5; %W/(m*°C)
retirado:ITE 50

    %Adobe, taipa, blocos de terra comprimida

        material.inertes_solos_terras(3).nome = 'Adobe, taipa, blocos de terra
comprimida';
        material.inertes_solos_terras(3).classificacao = 'Inertes, solos e
terras';
        material.inertes_solos_terras(3).rho = 1885; %kg/m^3
retirado:ITE 50
        material.inertes_solos_terras(3).k = 1.1; %W/(m*°C)
retirado:ITE 50

```

```
%Base de dados de materiais
```

```
% Dados relativos a Isolantes Térmicos
```

```
global material
```

```
    %Lã Mineral
```

```
        %Lã de Rocha
```

```
        material.isolamento(1).nome = 'Lã de Rocha';  
        material.isolamento(1).classificacao = 'Isolante Térmico';  
        material.isolamento(1).rho = 27.5;                %kg/m^3   retirado:ITE✓
```

```
50
```

```
        material.isolamento(1).k = 0.045;                %W/(m*°C) retirado:ITE✓
```

```
50
```

```
        %Lã de Vidro
```

```
        material.isolamento(2).nome = 'Lã de Vidro';  
        material.isolamento(2).classificacao = 'Isolante Térmico';  
        material.isolamento(2).rho = 11.5 ;              %kg/m^3   retirado:ITE✓
```

```
50
```

```
        material.isolamento(2).k = 0.045;                %W/(m*°C) retirado:ITE✓
```

```
50
```

```
    %Aglomerado de cortiça expandida (ICB)
```

```
        material.isolamento(3).nome = 'Aglomerado de cortiça expandida (ICB)';  
        material.isolamento(3).classificacao = 'Isolante Térmico';  
        material.isolamento(3).rho = 115;                %kg/m^3   retirado:ITE✓
```

```
50
```

```
        material.isolamento(3).k = 0.045;                %W/(m*°C) retirado:ITE✓
```

```
50
```

```
    %Aglomerado de cortiça natural com ligantes betuminosos ou sintéticos
```

```
        material.isolamento(4).nome = 'Aglomerado de cortiça natural com✓  
ligantes betuminosos ou sintéticos';  
        material.isolamento(4).classificacao = 'Isolante Térmico';  
        material.isolamento(4).rho = 125;                %kg/m^3   retirado:ITE✓
```

```
50
```

```
        material.isolamento(4).k = 0.050;                %W/(m*°C) retirado:ITE✓
```

```
50
```

```
    %Poliestireno expandido moldado (EPS)
```

```
        material.isolamento(5).nome = 'Aglomerado de cortiça natural com✓  
ligantes betuminosos ou sintéticos';  
        material.isolamento(5).classificacao = 'Isolante Térmico';  
        material.isolamento(5).rho = 10 ;                %kg/m^3   retirado:ITE✓
```

```
50
```

```
        material.isolamento(5).k = 0.055;                %W/(m*°C) retirado:ITE✓
```

```
50
```

```
    %Poliestireno expandido extrudido (XPS)
```

```
        material.isolamento(6).nome = 'Poliestireno expandido extrudido (XPS)';  
        material.isolamento(6).classificacao = 'Isolante Térmico';  
        material.isolamento(6).rho = 32.5 ;              %kg/m^3   retirado:ITE✓
```

```

50         material.isolamento(6).k = 0.037;                                %W/(m*°C) retirado:ITE✓
50
    %Espuma rígida de poliuretano(PUR) ou de poli-isocianurato (PIR) - em placas
        material.isolamento(7).nome = 'Espuma rígida de poliuretano(PUR) ou de✓
poli-isocianurato (PIR) - em placas ';
        material.isolamento(7).classificacao = 'Isolante Térmico';
        material.isolamento(7).rho = 35 ;                                %kg/m^3 retirado:ITE✓
50
        material.isolamento(7).k = 0.040;                                %W/(m*°C) retirado:ITE✓
50
    %Espuma rígida de poliuretano(PUR) ou de poli-isocianurato (PIR) - projectado✓
ou injectado in situ
        material.isolamento(8).nome = 'Espuma rígida de poliuretano(PUR) ou de✓
poli-isocianurato (PIR) - projectado ou injectado in situ ';
        material.isolamento(8).classificacao = 'Isolante Térmico';
        material.isolamento(8).rho = 35 ;                                %kg/m^3 retirado:ITE✓
50
        material.isolamento(8).k = 0.042;                                %W/(m*°C) retirado:ITE✓
50
    %Espuma rígida de poliuretano(PUR) ou de poli-isocianurato (PIR) - projectado✓
ou injectado in situ
        material.isolamento(9).nome = 'Espuma rígida de poliuretano(PUR) ou de✓
poli-isocianurato (PIR) - entre paramentos metálicos (painéis sanduíche) ';
        material.isolamento(9).classificacao = 'Isolante Térmico';
        material.isolamento(9).rho = 35 ;                                %kg/m^3 retirado:ITE✓
50
        material.isolamento(9).k = 0.042;                                %W/(m*°C) retirado:ITE✓
50
    %Espuma de polietileno expandido extrudido (PEF)
        material.isolamento(10).nome = 'Espuma de polietileno expandido✓
extrudido (PEF)';
        material.isolamento(10).classificacao = 'Isolante Térmico';
        material.isolamento(10).rho = 35 ;                                %kg/m^3 retirado:ITE✓
50
        material.isolamento(10).k = 0.050;                                %W/(m*°C) retirado:ITE✓
50
    %Grânulos leves ou fibras soltas (sem ligante)
        % Grânulos de argila, de vermiculite ou de perlite expandidas
        material.isolamento(11).nome = 'Grânulos de argila, de vermiculite ou de✓
perlite expandidas';
        material.isolamento(11).classificacao = 'Isolante Térmico';
        material.isolamento(11).rho = 350 ;                                %kg/m^3 retirado:ITE✓
50
        material.isolamento(11).k = 0.16;                                %W/(m*°C) retirado:ITE✓
50
        % Outros tipos de grânulos leves ou de fibras soltas

```

```

material.isolamento(12).nome = 'Outros tipos de grânulos leves ou de
fibras soltas';
material.isolamento(12).classificacao = 'Isolante Térmico';
material.isolamento(12).rho = 60 ;                %kg/m^3 retirado:ITE✓
50
material.isolamento(12).k = 0.060;                %W/(m*°C) retirado:ITE✓
50

% Espuma elastomérica flexível (FEF)

material.isolamento(13).nome = 'Espuma elastomérica flexível (FEF)';
material.isolamento(13).classificacao = 'Isolante Térmico';
material.isolamento(13).rho = 70 ;                %kg/m^3 retirado:ITE✓
50
material.isolamento(13).k = 0.050;                %W/(m*°C) retirado:ITE✓
50

```

```
%Base de dados de materiais
```

```
% Dados relativos a Madeiras e derivados
```

```
global material
```

```
%Madeiras maciças
```

```
%Balsa
```

```
material.madeira(1).nome = 'Madeira maciças - Balsa';  
material.madeira(1).classificacao = 'Madeira e derivados';  
material.madeira(1).rho = 150; %kg/m^3 retirado:ITE 50  
material.madeira(1).k = 0.057; %W/(m*°C) retirado:ITE 50
```

```
%Madeiras muito leves
```

```
material.madeira(2).nome = 'Madeira maciças - muito leves, excepto Balsa';  
material.madeira(2).classificacao = 'Madeira e derivados';  
material.madeira(2).rho = 320; %kg/m^3 retirado:ITE 50  
material.madeira(2).k = 0.13; %W/(m*°C) retirado:ITE 50
```

```
%Madeiras leves
```

```
material.madeira(3).nome = 'Madeira maciças - leves';  
material.madeira(3).classificacao = 'Madeira e derivados';  
material.madeira(3).rho = 500; %kg/m^3 retirado:ITE 50  
material.madeira(3).k = 0.15; %W/(m*°C) retirado:ITE 50
```

```
%Madeiras semi-densas
```

```
material.madeira(4).nome = 'Madeira maciças - semi-densas';  
material.madeira(4).classificacao = 'Madeira e derivados';  
material.madeira(4).rho = 660; %kg/m^3 retirado:ITE 50  
material.madeira(4).k = 0.18; %W/(m*°C) retirado:ITE 50
```

```
%Madeiras densas
```

```
material.madeira(5).nome = 'Madeira maciças - densas';  
material.madeira(5).classificacao = 'Madeira e derivados';  
material.madeira(5).rho = 810; %kg/m^3 retirado:ITE 50  
material.madeira(5).k = 0.23; %W/(m*°C) retirado:ITE 50
```

```
%Madeiras muito densas
```

```
material.madeira(6).nome = 'Madeira maciças - muito densas';  
material.madeira(6).classificacao = 'Madeira e derivados';  
material.madeira(6).rho = 900; %kg/m^3 retirado:ITE 50  
material.madeira(6).k = 0.29; %W/(m*°C) retirado:ITE 50
```

```
%Madeiras resinosas leves
```

```
material.madeira(7).nome = 'Madeira maciças - resinosas leves';  
material.madeira(7).classificacao = 'Madeira e derivados';  
material.madeira(7).rho = 400; %kg/m^3 retirado:ITE 50  
material.madeira(7).k = 0.13; %W/(m*°C) retirado:ITE 50
```

```
%Madeiras resinosas semi-densas
```

```
material.madeira(8).nome = 'Madeira maciças - resinosas semi-densas';
material.madeira(8).classificacao = 'Madeira e derivados';
material.madeira(8).rho = 480; %kg/m^3 retirado:ITE 50
material.madeira(8).k = 0.15; %W/(m*°C) retirado:ITE 50
```

%Madeiras resinosas densas

```
material.madeira(9).nome = 'Madeira maciças - resinosas densas';
material.madeira(9).classificacao = 'Madeira e derivados';
material.madeira(9).rho = 565; %kg/m^3 retirado:ITE 50
material.madeira(9).k = 0.18; %W/(m*°C) retirado:ITE 50
```

%Madeiras resinosas muito densas

```
material.madeira(10).nome = 'Madeira maciças - resinosas muito densas';
material.madeira(10).classificacao = 'Madeira e derivados';
material.madeira(10).rho = 650; %kg/m^3 retirado:ITE 50
material.madeira(10).k = 0.23; %W/(m*°C) retirado:ITE 50
```

%Painéis de contraplacado

```
material.madeira(11).nome = 'Painéis de contraplacado';
material.madeira(11).classificacao = 'Madeira e derivados';
material.madeira(11).rho = 300; %kg/m^3 retirado:ITE 50
material.madeira(11).k = 0.09; %W/(m*°C) retirado:ITE 50
```

%Painéis de aglomerado de partículas de madeira

```
material.madeira(12).nome = 'Painéis de aglomerado de partículas de
madeira';
material.madeira(12).classificacao = 'Madeira e derivados';
material.madeira(12).rho = 300; %kg/m^3 retirado:ITE 50
material.madeira(12).k = 0.10; %W/(m*°C) retirado:ITE 50
```

%Painéis de fibras de madeira (incluindo MDF)

```
material.madeira(13).nome = 'Painéis de fibras de madeira (incluindo
MDF)';
material.madeira(13).classificacao = 'Madeira e derivados';
material.madeira(13).rho = 250; %kg/m^3 retirado:ITE 50
material.madeira(13).k = 0.07; %W/(m*°C) retirado:ITE 50
```

%Painéis de lamelas longas orientadas (OSB)

```
material.madeira(14).nome = 'Painéis de lamelas longas orientadas
(OSB)';
material.madeira(14).classificacao = 'Madeira e derivados';
material.madeira(14).rho = 600; %kg/m^3 retirado:ITE 50
material.madeira(14).k = 0.13; %W/(m*°C) retirado:ITE 50
```

%Painéis de partículas de madeira aglomeradas com cimento (EN 634)

```
material.madeira(15).nome = 'Painéis de partículas de madeira
aglomeradas com cimento (EN 634)';
material.madeira(15).classificacao = 'Madeira e derivados';
material.madeira(15).rho = 1100; %kg/m^3 retirado:ITE 50
material.madeira(15).k = 0.23; %W/(m*°C) retirado:ITE 50
```

```
%Painéis de fibras de madeira aglomeradas com um ligante hidráulico("Lã de✓  
madeira")
```

```
material.madeira(16).nome = 'Painéis de fibras de madeira aglomeradas✓  
com um ligante hidráulico("Lã de madeira)';  
material.madeira(16).classificacao = 'Madeira e derivados';  
material.madeira(16).rho = 300;           %kg/m^3   retirado:ITE 50  
material.madeira(16).k = 0.10;           %W/(m*°C) retirado:ITE 50
```

```
%Base de dados de materiais
```

```
% Dados relativos a Materiais Plásticos (materiais sólidos não-expandidos)
```

```
global material
```

```
%Acrílicos
```

```
material.plasticos(1).nome = 'Acrílicos';  
material.plasticos(1).classificacao = 'Materiais Plásticos (materiais  
sólidos não-expandidos)';  
material.plasticos(1).rho = 1050; %kg/m^3 retirado:ITE 50  
material.plasticos(1).k = 0.20; %W/(m*°C) retirado:ITE 50
```

```
%Polimetacrilato de metilo (PMMA)
```

```
material.plasticos(2).nome = 'Polimetacrilato de metilo (PMMA)';  
material.plasticos(2).classificacao = 'Materiais Plásticos (materiais  
sólidos não-expandidos)';  
material.plasticos(2).rho = 1180; %kg/m^3 retirado:ITE 50  
material.plasticos(2).k = 0.18; %W/(m*°C) retirado:ITE 50
```

```
%Policarbonato
```

```
material.plasticos(3).nome = 'Policarbonato';  
material.plasticos(3).classificacao = 'Materiais Plásticos (materiais  
sólidos não-expandidos)';  
material.plasticos(3).rho = 1200; %kg/m^3 retirado:ITE 50  
material.plasticos(3).k = 0.20; %W/(m*°C) retirado:ITE 50
```

```
%Cloreto de polivinilo (PVC)
```

```
material.plasticos(4).nome = 'Cloreto de polivinilo (PVC)';  
material.plasticos(4).classificacao = 'Materiais Plásticos (materiais  
sólidos não-expandidos)';  
material.plasticos(4).rho = 1390; %kg/m^3 retirado:ITE 50  
material.plasticos(4).k = 0.17; %W/(m*°C) retirado:ITE 50
```

```
%Polietileno de baixa densidade
```

```
material.plasticos(5).nome = 'Polietileno de baixa densidade';  
material.plasticos(5).classificacao = 'Materiais Plásticos (materiais  
sólidos não-expandidos)';  
material.plasticos(5).rho = 920; %kg/m^3 retirado:ITE 50  
material.plasticos(5).k = 0.33; %W/(m*°C) retirado:ITE 50
```

```
%Polietileno de alta densidade
```

```
material.plasticos(6).nome = 'Polietileno de alta densidade';  
material.plasticos(6).classificacao = 'Materiais Plásticos (materiais  
sólidos não-expandidos)';  
material.plasticos(6).rho = 980; %kg/m^3 retirado:ITE 50  
material.plasticos(6).k = 0.50; %W/(m*°C) retirado:ITE 50
```

```
%Poliamida (Nylon)
```

```
material.plasticos(7).nome = 'Poliamida (Nylon)';  
material.plasticos(7).classificacao = 'Materiais Plásticos (materiais  
sólidos não-expandidos)';  
material.plasticos(7).rho = 1150; %kg/m^3 retirado:ITE 50  
material.plasticos(7).k = 0.25; %W/(m*°C) retirado:ITE 50
```



%Poliamida com 25% de fibra de vidro

```
material.plasticos(8).nome = 'Poliamida com 25% de fibra de vidro';
material.plasticos(8).classificacao = 'Materiais Plásticos (materiais
sólidos não-expandidos)';
material.plasticos(8).rho = 1450; %kg/m^3 retirado:ITE 50
material.plasticos(8).k = 0.30; %W/(m*°C) retirado:ITE 50
```

%Polipropileno

```
material.plasticos(9).nome = 'Polipropileno';
material.plasticos(9).classificacao = 'Materiais Plásticos (materiais
sólidos não-expandidos)';
material.plasticos(9).rho = 910; %kg/m^3 retirado:ITE 50
material.plasticos(9).k = 0.22; %W/(m*°C) retirado:ITE 50
```

%Polipropileno com 25% de fibra de vidro

```
material.plasticos(10).nome = 'Polipropileno com 25% de fibra de vidro';
material.plasticos(10).classificacao = 'Materiais Plásticos (materiais
sólidos não-expandidos)';
material.plasticos(10).rho = 1200; %kg/m^3 retirado:ITE 50
material.plasticos(10).k = 0.25; %W/(m*°C) retirado:ITE 50
```

%Resina de poliester

```
material.plasticos(11).nome = 'Resina de poliester';
material.plasticos(11).classificacao = 'Materiais Plásticos (materiais
sólidos não-expandidos)';
material.plasticos(11).rho = 1400; %kg/m^3 retirado:ITE 50
material.plasticos(11).k = 0.19; %W/(m*°C) retirado:ITE 50
```

%Resina epoxídica

```
material.plasticos(12).nome = 'Resina epoxídica';
material.plasticos(12).classificacao = 'Materiais Plásticos (materiais
sólidos não-expandidos)';
material.plasticos(12).rho = 1200; %kg/m^3 retirado:ITE 50
material.plasticos(12).k = 0.20; %W/(m*°C) retirado:ITE 50
```

%Resina fenólica

```
material.plasticos(13).nome = 'Resina fenólica';
material.plasticos(13).classificacao = 'Materiais Plásticos (materiais
sólidos não-expandidos)';
material.plasticos(13).rho = 1300; %kg/m^3 retirado:ITE 50
material.plasticos(13).k = 0.30; %W/(m*°C) retirado:ITE 50
```

%Uretano/poliuretano (perfis de corte térmico para caixilhos)

```
material.plasticos(14).nome = 'Uretano/poliuretano (perfis de corte
térmico para caixilhos)';
material.plasticos(14).classificacao = 'Materiais Plásticos (materiais
sólidos não-expandidos)';
material.plasticos(14).rho = 1300; %kg/m^3 retirado:ITE 50
material.plasticos(14).k = 0.21; %W/(m*°C) retirado:ITE 50
```

```
%Base de dados de materiais
```

```
% Dados relativos a Material Cerâmico
```

```
global material
```

```
    %Material cerâmico para tijolos, blocos, telhas e ladrilhos
```

```
        material.ceramico(1).nome = 'Material cerâmico para tijolos, blocos, telhas e ladrilhos';
```

```
        material.ceramico(1).classificacao = 'Material Cerâmico';
```

```
        material.ceramico(1).rho = 900;          %kg/m^3 retirado:ITE 50
```

```
        material.ceramico(1).k = 0.34;          %W/(m*°C) retirado:ITE 50
```

```
%Base de dados de materiais
```

```
% Dados relativos a Metais
```

```
global material
```

```
%Aço
```

```
material.metais(1).nome = 'Aço';  
material.metais(1).classificacao = 'Metais';  
material.metais(1).rho = 7800; %kg/m^3 retirado:ITE 50  
material.metais(1).k = 50; %W/(m*°C) retirado:ITE 50
```

```
%Aço inoxidável
```

```
material.metais(2).nome = 'Aço inoxidável';  
material.metais(2).classificacao = 'Metais';  
material.metais(2).rho = 7900; %kg/m^3 retirado:ITE 50  
material.metais(2).k = 17; %W/(m*°C) retirado:ITE 50
```

```
%Alumínio
```

```
material.metais(3).nome = 'Alumínio';  
material.metais(3).classificacao = 'Metais';  
material.metais(3).rho = 2700; %kg/m^3 retirado:ITE 50  
material.metais(3).k = 230; %W/(m*°C) retirado:ITE 50
```

```
%Ligas de alumínio
```

```
material.metais(4).nome = 'Ligas de alumínio';  
material.metais(4).classificacao = 'Metais';  
material.metais(4).rho = 2800; %kg/m^3 retirado:ITE 50  
material.metais(4).k = 160; %W/(m*°C) retirado:ITE 50
```

```
%Bronze
```

```
material.metais(5).nome = 'Bronze';  
material.metais(5).classificacao = 'Metais';  
material.metais(5).rho = 8700; %kg/m^3 retirado:ITE 50  
material.metais(5).k = 65; %W/(m*°C) retirado:ITE 50
```

```
%Chumbo
```

```
material.metais(6).nome = 'Chumbo';  
material.metais(6).classificacao = 'Metais';  
material.metais(6).rho = 11300; %kg/m^3 retirado:ITE 50  
material.metais(6).k = 35; %W/(m*°C) retirado:ITE 50
```

```
%Cobre
```

```
material.metais(7).nome = 'Cobre';  
material.metais(7).classificacao = 'Metais';  
material.metais(7).rho = 8900; %kg/m^3 retirado:ITE 50  
material.metais(7).k = 380; %W/(m*°C) retirado:ITE 50
```

```
%Ferro
```

```
material.metais(8).nome = 'Ferro';  
material.metais(8).classificacao = 'Metais';  
material.metais(8).rho = 7870; %kg/m^3 retirado:ITE 50
```

```
material.metalis(8).k = 72; %W/(m*°C) retirado:ITE 50

%Ferro fundido

material.metalis(9).nome = 'Ferro fundido';
material.metalis(9).classificacao = 'Metais';
material.metalis(9).rho = 7870; %kg/m^3 retirado:ITE 50
material.metalis(9).k = 72; %W/(m*°C) retirado:ITE 50

%Latão

material.metalis(10).nome = 'Latão';
material.metalis(10).classificacao = 'Metais';
material.metalis(10).rho = 8400; %kg/m^3 retirado:ITE 50
material.metalis(10).k = 120; %W/(m*°C) retirado:ITE 50

%Zinco

material.metalis(11).nome = 'Zinco';
material.metalis(11).classificacao = 'Metais';
material.metalis(11).rho = 7200; %kg/m^3 retirado:ITE 50
material.metalis(11).k = 110; %W/(m*°C) retirado:ITE 50
```

```
%Base de dados de materiais
```

```
% Dados relativos a Pedras Naturais
```

```
global material
```

```
%Rochas plutônicas e metamórficas
```

```
%Gneisse
```

```
material.pedras(1).nome = 'Gneisse';  
material.pedras(1).classificacao = 'Rochas plutônicas e metamórficas';  
material.pedras(1).rho = 2550; %kg/m^3 retirado:ITE 50  
material.pedras(1).k = 3.5; %W/(m*°C) retirado:ITE 50
```

```
%Granito
```

```
material.pedras(2).nome = 'Granito';  
material.pedras(2).classificacao = 'Rochas plutônicas e metamórficas';  
material.pedras(2).rho = 2600; %kg/m^3 retirado:ITE 50  
material.pedras(2).k = 2.8; %W/(m*°C) retirado:ITE 50
```

```
%Xisto, Ardósia (em paredes,fluxo de calor paralelo aos estratos)
```

```
material.pedras(3).nome = 'Xisto e Ardósia (em paredes,fluxo de calor  
paralelo aos estratos)';  
material.pedras(3).classificacao = 'Rochas plutônicas e metamórficas';  
material.pedras(3).rho = 2400; %kg/m^3 retirado:ITE 50  
material.pedras(3).k = 2.2; %W/(m*°C) retirado:ITE 50
```

```
%Rochas vulcânicas
```

```
%Basalto
```

```
material.pedras(4).nome = 'Basalto';  
material.pedras(4).classificacao = 'Rochas vulcânicas';  
material.pedras(4).rho = 2850; %kg/m^3 retirado:ITE 50  
material.pedras(4).k = 1.1; %W/(m*°C) retirado:ITE 50
```

```
%Traquito e Andesito
```

```
material.pedras(5).nome = 'Tranquito e Andesito';  
material.pedras(5).classificacao = 'Rochas vulcânicas';  
material.pedras(5).rho = 2350; %kg/m^3 retirado:ITE 50  
material.pedras(5).k = 1.1; %W/(m*°C) retirado:ITE 50
```

```
%Rochas porosas
```

```
material.pedras(6).nome = 'Rochas porosas';  
material.pedras(6).classificacao = 'Rochas vulcânicas';  
material.pedras(6).rho = 1500; %kg/m^3 retirado:ITE 50  
material.pedras(6).k = 0.55; %W/(m*°C) retirado:ITE 50
```

```
%Pedra-pomes
```

```
material.pedras(7).nome = 'Pedra-pomes';  
material.pedras(7).classificacao = 'Rochas vulcânicas';  
material.pedras(7).rho = 300; %kg/m^3 retirado:ITE 50  
material.pedras(7).k = 0.12; %W/(m*°C) retirado:ITE 50
```

%Rochas calcárias

%Mármore

```
material.pedras(8).nome = 'Mármore';
material.pedras(8).classificacao = 'Rochas calcárias';
material.pedras(8).rho = 2700; %kg/m^3 retirado:ITE 50
material.pedras(8).k = 3.5; %W/(m*°C) retirado:ITE 50
```

%Pedras calcárias muito duras

```
material.pedras(9).nome = 'Pedras calcárias muito duras';
material.pedras(9).classificacao = 'Rochas calcárias';
material.pedras(9).rho = 2395; %kg/m^3 retirado:ITE 50
material.pedras(9).k = 2.3; %W/(m*°C) retirado:ITE 50
```

%Pedras calcárias duras

```
material.pedras(10).nome = 'Pedras calcárias duras';
material.pedras(10).classificacao = 'Rochas calcárias';
material.pedras(10).rho = 2095; %kg/m^3 retirado:ITE 50
material.pedras(10).k = 1.7; %W/(m*°C) retirado:ITE 50
```

%Pedras calcárias densas

```
material.pedras(11).nome = 'Pedras calcárias densas';
material.pedras(11).classificacao = 'Rochas calcárias';
material.pedras(11).rho = 1895; %kg/m^3 retirado:ITE 50
material.pedras(11).k = 1.4; %W/(m*°C) retirado:ITE 50
```

%Pedras calcárias macias

```
material.pedras(12).nome = 'Pedras calcárias macias';
material.pedras(12).classificacao = 'Rochas calcárias';
material.pedras(12).rho = 1695; %kg/m^3 retirado:ITE 50
material.pedras(12).k = 1.1; %W/(m*°C) retirado:ITE 50
```

%Pedras calcárias muito macias

```
material.pedras(13).nome = 'Pedras calcárias muito macias';
material.pedras(13).classificacao = 'Rochas calcárias';
material.pedras(13).rho = 1500; %kg/m^3 retirado:ITE 50
material.pedras(13).k = 0.85; %W/(m*°C) retirado:ITE 50
```

%Grés

%Grés quartzoso

```
material.pedras(14).nome = 'Grés quartzoso';
material.pedras(14).classificacao = 'Grés ';
material.pedras(14).rho = 2700; %kg/m^3 retirado:ITE 50
material.pedras(14).k = 2.6; %W/(m*°C) retirado:ITE 50
```

%Grés silicioso

```
material.pedras(15).nome = 'Grés silicioso';
material.pedras(15).classificacao = 'Grés ';
material.pedras(15).rho = 2395; %kg/m^3 retirado:ITE 50
material.pedras(15).k = 2.3; %W/(m*°C) retirado:ITE 50
```

```
%Grés calcário
```

```
material.pedras(16).nome = 'Grés quartzoso';  
material.pedras(16).classificacao = 'Grés ';  
material.pedras(16).rho = 2350; %kg/m^3 retirado:ITE 50  
material.pedras(16).k = 1.9; %W/(m*°C) retirado:ITE 50
```

```
%Silex
```

```
%Silex
```

```
material.pedras(17).nome = 'Silex';  
material.pedras(17).classificacao = 'Silex ';  
material.pedras(17).rho = 2700; %kg/m^3 retirado:ITE 50  
material.pedras(17).k = 2.6; %W/(m*°C) retirado:ITE 50
```

```
% Dados relativos a Pedras artificiais
```

```
%Pedras artificiais
```

```
material.pedras(18).nome = 'Pedras artificiais (incluindo juntas de✓  
assentamento)';  
material.pedras(18).classificacao = 'Pedras artificiais ';  
material.pedras(18).rho = 1750; %kg/m^3 retirado:ITE 50  
material.pedras(18).k = 1.3; %W/(m*°C) retirado:ITE 50
```

```
% Dados relativos a Revestimentos de pisos ou de paredes
global material
```

```
material.revestimentos_pisos_paredes(1).nome = 'Ladrilhos ou rolos :↵
borracha';
material.revestimentos_pisos_paredes(1).classificacao = 'Revestimentos↵
de pisos ou de paredes';
material.revestimentos_pisos_paredes(1).rho = 1200; %↵
kg/m^3 retirado:ITE 50
material.revestimentos_pisos_paredes(1).k = 0.17; %W/↵
(m*°C) retirado:ITE 50
```

```
material.revestimentos_pisos_paredes(2).nome = 'Ladrilhos ou rolos :  
plástico';  
material.revestimentos_pisos_paredes(2).classificacao = 'Revestimentos  
de pisos ou de paredes';  
material.revestimentos_pisos_paredes(2).rho = 1000; %  
kg/m^3 retirado:ITE 50  
material.revestimentos_pisos_paredes(2).k = 0.20; %W/  
(m*°C) retirado:ITE 50
```

```
material.revestimentos_pisos_paredes(3).nome = 'Ladrilhos ou rolos :  
aglomerado de cortiça';  
material.revestimentos_pisos_paredes(3).classificacao = 'Revestimentos  
de pisos ou de paredes';  
material.revestimentos_pisos_paredes(3).rho = 500; %  
kg/m^3 retirado:ITE 50  
material.revestimentos_pisos_paredes(3).k = 0.065; %W/  
(m*°C) retirado:ITE 50
```

```
material.revestimentos_pisos_paredes(4).nome = 'Ladrilhos ou rolos :↵
revestimento têxtil (carpete, alcatifa)';
material.revestimentos_pisos_paredes(4).classificacao = 'Revestimentos↵
de pisos ou de paredes';
material.revestimentos_pisos_paredes(4).rho = 200; %↵
kg/m^3 retirado:ITE 50
material.revestimentos_pisos_paredes(4).k = 0.060; %W/↵
(m*°C) retirado:ITE 50
```

```
material.revestimentos_pisos_paredes(5).nome = 'Ladrilhos ou rolos :↵
linóleo';
material.revestimentos_pisos_paredes(5).classificacao = 'Revestimentos↵
de pisos ou de paredes';
material.revestimentos_pisos_paredes(5).rho = 1200; %↵
kg/m^3 retirado:ITE 50
material.revestimentos_pisos_paredes(5).k = 0.17; %W/↵
(m*°C) retirado:ITE 50
```



%Ladrilhos ou rolos : cerâmica vidrada/grés cerâmico

```
material.revestimentos_pisos_paredes(6).nome = 'Ladrilhos ou rolos :  
cerâmica vidrada/grés cerâmico';  
material.revestimentos_pisos_paredes(6).classificacao = 'Revestimentos  
de pisos ou de paredes';  
material.revestimentos_pisos_paredes(6).rho = 2300; %  
kg/m^3 retirado:ITE 50  
material.revestimentos_pisos_paredes(6).k = 1.3; %W/  
(m*°C) retirado:ITE 50
```

%Subcamada de feltro

```
material.revestimentos_pisos_paredes(7).nome = 'Subcamada de feltro';
material.revestimentos_pisos_paredes(7).classificacao = 'Revestimentos
de pisos ou de paredes';
material.revestimentos_pisos_paredes(7).rho = 120; % kg/m^3 retirado:ITE 50
material.revestimentos_pisos_paredes(7).k = 0.050; % W/(m*°C) retirado:ITE 50
```

%Subcamada de aglomerado de cortiça

```
material.revestimentos_pisos_paredes(8).nome = 'Subcamada de aglomerado
de cortiça';
material.revestimentos_pisos_paredes(8).classificacao = 'Revestimentos
de pisos ou de paredes';
material.revestimentos_pisos_paredes(8).rho = 150; %
kg/m^3 retirado:ITE 50
material.revestimentos_pisos_paredes(8).k = 0.050; %W/(
(m*°C) retirado:ITE 50
```

%Subcamada de espuma de borracha ou de plástico celular

```
material.revestimentos_pisos_paredes(9).nome = 'Subcamada de espuma de
borracha ou de plástico celular';
material.revestimentos_pisos_paredes(9).classificacao = 'Revestimentos
de pisos ou de paredes';
material.revestimentos_pisos_paredes(9).rho = 270; %
kg/m^3 retirado:ITE 50
material.revestimentos_pisos_paredes(9).k = 0.10; %W/
(m*°C) retirado:ITE 50
```

%Subcamada de lâ

```
material.revestimentos_pisos_paredes(10).nome = 'Subcamada de espuma de
borracha ou de plástico celular';
material.revestimentos_pisos_paredes(10).classificacao = 'Revestimentos
de pisos ou de paredes';
material.revestimentos_pisos_paredes(10).rho = 200; %
kg/m^3 retirado:ITE 50
material.revestimentos_pisos_paredes(10).k = 0.060; %W/
(m*°C) retirado:ITE 50
```

```
%Base de dados de materiais
```

```
% Dados relativos a Vidros
```

```
global material
```

```
%Sódio-calcário (incluindo vidro float)
```

```
material.vidros(1).nome = 'Sódio-calcário (incluindo vidro float)';  
material.vidros(1).classificacao = 'Vidros';  
material.vidros(1).rho = 2500; %kg/m^3 retirado:ITE 50  
material.vidros(1).k = 1.00; %W/(m*°C) retirado:ITE 50
```

```
%Vidro de quartzo
```

```
material.vidros(2).nome = 'Vidro de quartzo';  
material.vidros(2).classificacao = 'Vidros';  
material.vidros(2).rho = 2200; %kg/m^3 retirado:ITE 50  
material.vidros(2).k = 1.40; %W/(m*°C) retirado:ITE 50
```

```
%Mosaico de vidro
```

```
material.vidros(3).nome = 'Mosaico de vidro';  
material.vidros(3).classificacao = 'Vidros';  
material.vidros(3).rho = 2000; %kg/m^3 retirado:ITE 50  
material.vidros(3).k = 1.20; %W/(m*°C) retirado:ITE 50
```

```
% Base de dados de elementos construtivos da envolvente
```

```
% Paredes de Fachada simples sem isolamento térmico
```

```
elementos(1).nome = 'Paredes simples de fachada sem isolamento térmico: tijolo furado';
```

```
elementos(1).tipo = 'Parede';
```

```
elementos(1).u = 1.3;
```

```
elementos(1).rsi = '-';
```

```
elementos(1).rse = '-';
```

```
elementos(1).tabela = {'-', 'Tijolo furado', '-', '-', '200 a 240'};
```

```
elementos(2).nome = 'Paredes simples de fachada sem isolamento térmico: blocos e betão normal';
```

```
elementos(2).tipo = 'Parede';
```

```
elementos(2).u = 1.9;
```

```
elementos(2).rsi = '-';
```

```
elementos(2).rse = '-';
```

```
elementos(2).tabela = {'-', 'Blocos de betão normal', '-', '-', '200 a 300'};
```

```
elementos(3).nome = 'Paredes simples de fachada sem isolamento térmico: blocos de betão leve';
```

```
elementos(3).tipo = 'Parede';
```

```
elementos(3).u = 1.3;
```

```
elementos(3).rsi = '-';
```

```
elementos(3).rse = '-';
```

```
elementos(3).tabela = {'-', 'Blocos de betão leve', '-', '-', '200 a 300'};
```

```
elementos(4).nome = 'Paredes simples de fachada sem isolamento térmico: pedra';
```

```
elementos(4).tipo = 'Parede';
```

```
elementos(4).u = 2.9;
```

```
elementos(4).rsi = '-';
```

```
elementos(4).rse = '-';
```

```
elementos(4).tabela = {'-', 'Pedra', '-', '-', '400 a 600'};
```

```
elementos(5).nome = 'Paredes simples de fachada sem isolamento térmico: parede de betão';
```

```
elementos(5).tipo = 'Parede';
```

```
elementos(5).u = 3.6;
```

```
elementos(5).rsi = '-';
```

```
elementos(5).rse = '-';
```

```
elementos(5).tabela = {'-', 'Parede de Betão', '-', '-', '100 a 200'};
```

```
% Paredes de Fachada simples com isolamento térmico pelo exterior
```

```
elementos(6).nome = 'Paredes simples de fachada com isolamento térmico(XPS) pelo exterior: tijolo furado';
```

```
elementos(6).tipo = 'Parede';
```

```
elementos(6).u = 0.67;
```

```
elementos(6).rsi = '-';
```

```
elementos(6).rse = '-';
```

```
elementos(6).tabela = {'-', 'Tijolo furado', '-', '-', '200 a 240'; '-', 'XPS', '25-40', '0.037', '30'};
```

```
elementos(7).nome = 'Paredes simples de fachada com isolamento térmico(XPS) pelo exterior: blocos e betão normal';
```

```
elementos(7).tipo = 'Parede';
```

```
elementos(7).u = 0.76;
```

```
elementos(7).rsi = '-';
```

```
elementos(7).rse = '-';
```

```
elementos(7).tabela = {'-', 'Blocos de betão normal', '-', '-', '200 a 300'; '-', 'XPS', '25-40', '0.037', '30'};
```

```
300';'-','XPS','25-40','0.037','30'};
```

```
elementos(8).nome = 'Paredes simples de fachada com isolamento térmico(XPS) pelo exterior: blocos de betão leve';  
elementos(8).tipo = 'Parede';  
elementos(8).u = 0.67;  
elementos(8).rsi = '-';  
elementos(8).rse = '-';  
elementos(8).tabela = {'-','Blocos de betão leve','-','-','200 a 300';'-','XPS','25-40','0.037','30'};
```

```
elementos(9).nome = 'Paredes simples de fachada com isolamento térmico(XPS) pelo exterior: pedra';  
elementos(9).tipo = 'Parede';  
elementos(9).u = 0.86;  
elementos(9).rsi = '-';  
elementos(9).rse = '-';  
elementos(9).tabela = {'-','Pedra','-','-','400 a 600';'-','XPS','25-40','0.037','30'};
```

```
elementos(10).nome = 'Paredes simples de fachada com isolamento térmico(XPS) pelo exterior: parede de betão';  
elementos(10).tipo = 'Parede';  
elementos(10).u = 0.92;  
elementos(10).rsi = '-';  
elementos(10).rse = '-';  
elementos(10).tabela = {'-','Parede de Betão','-','-','100 a 200';'-','XPS','25-40','0.037','30'};
```

```
% Paredes de Fachada com isolamento térmico pelo interior
```

```
elementos(11).nome = 'Paredes simples de fachada com isolamento térmico(XPS) pelo interior: tijolo furado';  
elementos(11).tipo = 'Parede';  
elementos(11).u = 0.63;  
elementos(11).rsi = '-';  
elementos(11).rse = '-';  
elementos(11).tabela = {'-','Tijolo furado','-','-','200 a 240';'-','XPS','25-40','0.037','30'};
```

```
elementos(12).nome = 'Paredes simples de fachada com isolamento térmico(XPS) pelo interior: blocos e betão normal';  
elementos(12).tipo = 'Parede';  
elementos(12).u = 0.73;  
elementos(12).rsi = '-';  
elementos(12).rse = '-';  
elementos(12).tabela = {'-','Blocos de betão normal','-','-','200 a 300';'-','XPS','25-40','0.037','30'};
```

```
elementos(13).nome = 'Paredes simples de fachada com isolamento térmico(XPS) pelo interior: blocos de betão leve';  
elementos(13).tipo = 'Parede';  
elementos(13).u = 0.63;  
elementos(13).rsi = '-';  
elementos(13).rse = '-';  
elementos(13).tabela = {'-','Blocos de betão leve','-','-','200 a 300';'-','XPS','25-40','0.037','30'};
```

```
elementos(14).nome = 'Paredes simples de fachada com isolamento térmico(XPS) pelo interior: pedra';
```

```

elementos(14).tipo = 'Parede';
elementos(14).u = 0.83;
elementos(14).rsi = '-';
elementos(14).rse = '-';
elementos(14).tabela = {'-', 'Pedra', '-', '-', '400 a 600'; '-', 'XPS', '25-40', '0.037', '30'};

elementos(15).nome = 'Paredes simples de fachada com isolamento térmico(XPS) pelo interior: parede de betão';
elementos(15).tipo = 'Parede';
elementos(15).u = 0.89;
elementos(15).rsi = '-';
elementos(15).rse = '-';
elementos(15).tabela = {'-', 'Parede de Betão', '-', '-', '100 a 200'; '-', 'XPS', '25-40', '0.037', '30'};

% Pavimentos sobre espaços exteriores - Isolamento térmico pelo exterior

elementos(16).nome = 'Pavimentos sobre espaços exteriores - Isolamento térmico(XPS) pelo exterior: Laje maciça';
elementos(16).tipo = 'Pavimento';
elementos(16).u = 0.85;
elementos(16).rsi = '-';
elementos(16).rse = '-';
elementos(16).tabela = {'-', 'Laje maciça', '-', '-', '100 a 200'; '-', 'XPS', '25-40', '0.037', '30'};

elementos(17).nome = 'Pavimentos sobre espaços exteriores - Isolamento térmico(XPS) pelo exterior: Laje aligeirada, blocos cerâmicos';
elementos(17).tipo = 'Pavimento';
elementos(17).u = 0.81;
elementos(17).rsi = '-';
elementos(17).rse = '-';
elementos(17).tabela = {'-', 'Laje aligeirada, blocos cerâmicos', '-', '-', '130 a 150'; '-', 'XPS', '25-40', '0.037', '30'};

elementos(18).nome = 'Pavimentos sobre espaços exteriores - Isolamento térmico(XPS) pelo exterior: Laje aligeirada, blocos de betão normal';
elementos(18).tipo = 'Pavimento';
elementos(18).u = 0.82;
elementos(18).rsi = '-';
elementos(18).rse = '-';
elementos(18).tabela = {'-', 'Laje aligeirada, blocos de betão normal', '-', '-', '130 a 150'; '-', 'XPS', '25-40', '0.037', '30'};

elementos(19).nome = 'Pavimentos sobre espaços exteriores - Isolamento térmico(XPS) pelo exterior: Laje aligeirada, blocos de betão leve';
elementos(19).tipo = 'Pavimento';
elementos(19).u = 0.80;
elementos(19).rsi = '-';
elementos(19).rse = '-';
elementos(19).tabela = {'-', 'Parede de Betão', '-', '-', '130 a 150'; '-', 'XPS', '25-40', '0.037', '30'};

% Coberturas horizontais (em terraço), sem isolamento térmico

elementos(20).nome = 'Coberturas horizontais (em terraço), sem isolamento térmico: Laje maciça';
elementos(20).tipo = 'Cobertura';
elementos(20).u = 1.6;

```

```
elementos(20).rsi = '-';
elementos(20).rse = '-';
elementos(20).tabela = {'-', 'Laje maciça', '-', '-', '100 a 200'};

elementos(21).nome = 'Coberturas horizontais (em terraço), sem isolamento térmico:✓
Laje aligeirada, blocos cerâmicos';
elementos(21).tipo = 'Cobertura';
elementos(21).u = 1.4;
elementos(21).rsi = '-';
elementos(21).rse = '-';
elementos(21).tabela = {'-', 'Laje aligeirada, blocos cerâmicos', '-', '-', '130 a✓
150'};

elementos(22).nome = 'Coberturas horizontais (em terraço), sem isolamento térmico:✓
Laje aligeirada, blocos de betão normal';
elementos(22).tipo = 'Cobertura';
elementos(22).u = 1.5;
elementos(22).rsi = '-';
elementos(22).rse = '-';
elementos(22).tabela = {'-', 'Laje aligeirada, blocos de betão normal', '-', '-', '130 a✓
150'};

elementos(23).nome = 'Coberturas horizontais (em terraço), sem isolamento térmico:✓
Laje aligeirada, blocos de betão leve';
elementos(23).tipo = 'Cobertura';
elementos(23).u = 1.4;
elementos(23).rsi = '-';
elementos(23).rse = '-';
elementos(23).tabela = {'-', 'Parede de Betão', '-', '-', '130 a 150'};
```

```
% Dados referentes a Portugal

% Aveiro
clima(1).nome='Aveiro';
clima(1).latitude='40,6°N (nominal)';
clima(1).longitude='8,6°W (nominal)';
clima(1).altitude= '50';

% Beja
clima(2).nome='Beja';
clima(2).latitude='37,9°N (nominal)';
clima(2).longitude='7,9°W (nominal)';
clima(2).altitude= '178';

% Braga
clima(3).nome='Braga';
clima(3).latitude='41,6°N (nominal)';
clima(3).longitude='8,5°W (nominal)';
clima(3).altitude= '171';

% Bragança
clima(4).nome='Bragança';
clima(4).latitude='41,6°N (nominal)';
clima(4).longitude='7,1°W (nominal)';
clima(4).altitude= '680';

% Castelo Branco
clima(5).nome='Castelo Branco';
clima(5).latitude='39,9°N (nominal)';
clima(5).longitude='7,3°W (nominal)';
clima(5).altitude= '328';

% Coimbra
clima(6).nome='Coimbra';
clima(6).latitude='40,3°N (nominal)';
clima(6).longitude='8,6°W (nominal)';
clima(6).altitude= '67';

% Évora
clima(7).nome='Évora';
clima(7).latitude='38,6°N (nominal)';
clima(7).longitude='7,8°W (nominal)';
clima(7).altitude= '221';

% Faro
clima(8).nome='Faro';
clima(8).latitude='37,3°N (nominal)';
clima(8).longitude='8,0°W (nominal)';
clima(8).altitude= '145';

% Funchal
clima(9).nome='Funchal';
clima(9).latitude='32,7°N (nominal)';
clima(9).longitude='16,9°W (nominal)';
clima(9).altitude= '380';

% Guarda
clima(10).nome='Guarda';
clima(10).latitude='40,6°N (nominal)';
clima(10).longitude='7,1°W (nominal)';
clima(10).altitude= '717';

% Leiria
clima(11).nome='Leiria';
clima(11).latitude='39,8,°N (nominal)';
clima(11).longitude='8,8°W (nominal)';
clima(11).altitude= '126';

% Lisboa
clima(12).nome='Lisboa';
clima(12).latitude='38,9°N (nominal)';
```

```
clima(12).longitude='9,2°W (nominal)';
clima(12).altitude= '109';
% Ponta Delgada
clima(13).nome='Ponta Delgada';
clima(13).latitude='37,8°N (nominal)';
clima(13).longitude='27,8°W (nominal)';
clima(13).altitude= '10';
% Portalegre
clima(14).nome='Portalegre';
clima(14).latitude='39,1°N (nominal)';
clima(14).longitude='7,6°W (nominal)';
clima(14).altitude= '246';
% Porto
clima(15).nome='Porto';
clima(15).latitude='41,3°N (nominal)';
clima(15).longitude='8,6°W (nominal)';
clima(15).altitude= '94';
% Santarém
clima(16).nome='Santarém';
clima(16).latitude='39,1°N (nominal)';
clima(16).longitude='8,6°W (nominal)';
clima(16).altitude= '73';
% Setúbal
clima(17).nome='Setúbal';
clima(17).latitude='38,6°N (nominal)';
clima(17).longitude='8,9°W (nominal)';
clima(17).altitude= '47';
% Viana do Castelo
clima(18).nome='Viana do Castelo';
clima(18).latitude='41,9°N (nominal)';
clima(18).longitude='8,5°W (nominal)';
clima(18).altitude= '268';
% Vila Real
clima(19).nome='Vila Real';
clima(19).latitude='41,1°N (nominal)';
clima(19).longitude='7,4°W (nominal)';
clima(19).altitude= '579';
% Viseu
clima(20).nome='Viseu';
clima(20).latitude='40,7°N (nominal)';
clima(20).longitude='7,9°W (nominal)';
clima(20).altitude= '497';
```



```
% Base de dados de caixilhos
```

```
caixilho(1).nome = 'Caixilho de madeira (densidade = 700 kg/m^3 ,cond.térmica = 0.18✓  
W/(mk),espessura = 50 mm) ';  
caixilho(1).uf = '2'; %W/(m^2K)
```

```
caixilho(2).nome = 'Caixilho de madeira (densidade = 500 kg/m^3 ,cond.térmica = 0.13✓  
W/(mk),espessura = 50 mm) ';  
caixilho(2).uf = '2.4'; %W/(m^2K)
```

```
caixilho(3).nome = 'Caixilho em poliuretano (com núcleo em metal) ';  
caixilho(3).uf = '2.8'; %W/(m^2K)
```

```
caixilho(4).nome = 'PVC - com duas câmaras ocas (com uma distância de pelo menos 5 mm✓  
entre paredes das câmaras) ';  
caixilho(4).uf = '2.2'; %W/(m^2K)
```

```
caixilho(5).nome = 'PVC - com três câmaras ocas (com uma distância de pelo menos 5 mm✓  
entre paredes das câmaras) ';  
caixilho(5).uf = '2'; %W/(m^2K)
```

```
% Base de dados de janelas - ITE 50
```

```
% Caixilharia em Madeira
```

```
janela(1).nome='Vão envidraçado vertical com caixilharia em madeira(Janela fixa,✓  
giratória ou de correr com vidro simples)';  
janela(1).tipo='Janela simples';  
janela(1).uw='5.1'; %W/(m^2.K)  
  
janela(2).nome='Vão envidraçado vertical com caixilharia em madeira(Janela fixa,✓  
giratória ou de correr com vidro duplo, com espaçamento da lâmina de ar de 6 mm)';  
janela(2).tipo='Janela simples';  
janela(2).uw='3.3'; %W/(m^2.K)  
  
janela(3).nome='Vão envidraçado vertical com caixilharia em madeira(Janela fixa,✓  
giratória ou de correr com vidro duplo, com espaçamento da lâmina de ar de 16 mm)';  
janela(3).tipo='Janela simples';  
janela(3).uw='2.6'; %W/(m^2.K)  
  
janela(4).nome='Vão envidraçado vertical com caixilharia em madeira(Janela fixa,✓  
giratória ou de correr com vidro simples em cada janela, com uma distância entre✓  
janelas de 50 a 100 mm)';  
janela(4).tipo='Janela dupla';  
janela(4).uw='2.5'; %W/(m^2.K)
```

```
%Caixilharia em metal sem corte térmico
```

```
janela(5).nome='Vão envidraçado vertical com caixilharia em metal sem corte térmico✓  
(Janela fixa com vidro simples)';  
janela(5).tipo='Janela simples';  
janela(5).uw='6'; %W/(m^2.K)  
  
janela(6).nome='Vão envidraçado vertical com caixilharia em metal sem corte térmico✓  
(Janela giratória com vidro simples)';  
janela(6).tipo='Janela simples';  
janela(6).uw='6.2'; %W/(m^2.K)  
  
janela(7).nome='Vão envidraçado vertical com caixilharia em metal sem corte térmico✓  
(Janela de correr com vidro simples)';  
janela(7).tipo='Janela simples';  
janela(7).uw='6.5'; %W/(m^2.K)  
  
janela(8).nome='Vão envidraçado vertical com caixilharia em metal sem corte térmico✓  
(Janela fixa com vidro duplo, espaçamento da lâmina de ar de 6 mm)';  
janela(8).tipo='Janela simples';  
janela(8).uw='3.9'; %W/(m^2.K)  
  
janela(9).nome='Vão envidraçado vertical com caixilharia em metal sem corte térmico✓  
(Janela fixa com vidro duplo, espaçamento da lâmina de ar de 16 mm)';  
janela(9).tipo='Janela simples';  
janela(9).uw='3.5'; %W/(m^2.K)  
  
janela(10).nome='Vão envidraçado vertical com caixilharia em metal sem corte térmico✓  
(Janela giratória com vidro duplo, espaçamento da lâmina de ar de 6 mm)';  
janela(10).tipo='Janela simples';  
janela(10).uw='4.3'; %W/(m^2.K)  
  
janela(11).nome='Vão envidraçado vertical com caixilharia em metal sem corte térmico✓  
(Janela giratória com vidro duplo, espaçamento da lâmina de ar de 16 mm)';  
janela(11).tipo='Janela simples';  
janela(11).uw='3.8'; %W/(m^2.K)
```

```
janela(12).nome='Vão envidraçado vertical com caixilharia em metal sem corte térmico'✓  
(Janela de correr com vidro duplo, espaçamento da lâmina de ar de 6 mm)';  
janela(12).tipo='Janela simples';  
janela(12).uw='4.5'; %W/(m^2.K)
```

```
janela(13).nome='Vão envidraçado vertical com caixilharia em metal sem corte térmico'✓  
(Janela de correr com vidro duplo, espaçamento da lâmina de ar de 16 mm)';  
janela(13).tipo='Janela simples';  
janela(13).uw='4.0'; %W/(m^2.K)
```

```
janela(14).nome='Vão envidraçado vertical com caixilharia em metal sem corte térmico'✓  
(Janela fixa, giratória ou de correr.Com vidro simples em cada janela.Distância✓  
entre janelas de 50 a 100 mm)';  
janela(14).tipo='Janela dupla';  
janela(14).uw='3.1'; %W/(m^2.K)
```

%Caixilharia em metal com corte térmico

```
janela(15).nome='Vão envidraçado vertical com caixilharia em metal com corte térmico'✓  
(Janela fixa, giratória ou de correr com vidro simple)';  
janela(15).tipo='Janela simples';  
janela(15).uw='5.4'; %W/(m^2.K)
```

```
janela(16).nome='Vão envidraçado vertical com caixilharia em metal com corte térmico'✓  
(Janela fixa, giratória ou de correr com vidro duplo. Espaçamento da lâmina de ar de✓  
6 mm)';  
janela(16).tipo='Janela simples';  
janela(16).uw='3.7'; %W/(m^2.K)
```

```
janela(17).nome='Vão envidraçado vertical com caixilharia em metal com corte térmico'✓  
(Janela fixa, giratória ou de correr com vidro duplo. Espaçamento da lâmina de ar de✓  
16 mm)';  
janela(17).tipo='Janela simples';  
janela(17).uw='3.3'; %W/(m^2.K)
```

%Caixilharia de Plástico

```
janela(18).nome='Vão envidraçado vertical com caixilharia em plástico(Janela fixa,✓  
giratória ou de correr com vidro simple)';  
janela(18).tipo='Janela simples';  
janela(18).uw='4.9'; %W/(m^2.K)
```

```
janela(19).nome='Vão envidraçado vertical com caixilharia em plástico(Janela fixa,✓  
giratória ou de correr com vidro duplo. Espaçamento da lâmina de ar de 6 mm)';  
janela(19).tipo='Janela simples';  
janela(19).uw='3.2'; %W/(m^2.K)
```

```
janela(20).nome='Vão envidraçado vertical com caixilharia em plástico(Janela fixa,✓  
giratória ou de correr com vidro duplo. Espaçamento da lâmina de ar de 16 mm)';  
janela(20).tipo='Janela simples';  
janela(20).uw='2.7'; %W/(m^2.K)
```

```
janela(21).nome='Vão envidraçado vertical com caixilharia em plástico(Janela fixa,✓  
giratória ou de correr.Com vidro simples em cada janela.Distância entre janelas de✓  
50 a 100 mm)';  
janela(21).tipo='Janela dupla';  
janela(21).uw='2.4'; %W/(m^2.K)
```

```
% Base de dados de portas
```

```
% Retirado ASHRAE Fundamentals 2013
```

```
porta(1).nome='Porta giratória - 3 folhas (2.44 x 2.13 m)';
```

```
porta(1).ud=4.46;
```

```
porta(2).nome='Porta giratória - 4 folhas (2.13 x 1.98 m)';
```

```
porta(2).ud=3.56;
```

```
porta(3).nome='Porta de madeira com caixilho em madeira';
```

```
porta(3).ud=2.61;
```

```
porta(4).nome='Porta de aluminio com caixilho em aluminio e vidro simples - sem  
corte térmico';
```

```
porta(4).ud=7.49;
```

```
porta(5).nome='Porta de aluminio com caixilho em aluminio e vidro simples - com  
corte térmico';
```

```
porta(5).ud=6.42;
```

```
% Base de dados - Vidro
% Retirado de ASHRAE Handbook - Fundamentals SI Chapter 15.8

vidro(1).nome = 'Vidro simples:3 mm Vidro';
vidro(1).tipo = 'Vidro simples';
vidro(1).ug = 5.91; %W/(m^2.K)

vidro(2).nome = 'Vidro duplo:6 mm espaço de ar(3 mm de vidro)';
vidro(2).tipo = 'Vidro duplo';
vidro(2).ug = 3.12; %W/(m^2.K)

vidro(3).nome = 'Vidro duplo:13 mm espaço de ar (3 mm de vidro)';
vidro(3).tipo = 'Vidro duplo';
vidro(3).ug = 2.73; %W/(m^2.K)

vidro(4).nome = 'Vidro triplo:6 mm espaço de ar(3 mm de vidro)';
vidro(4).tipo = 'Vidro triplo';
vidro(4).ug = 2.16; %W/(m^2.K)

vidro(5).nome = 'Vidro triplo:13 mm espaço de ar(3 mm de vidro)';
vidro(5).tipo = 'Vidro triplo';
vidro(5).ug = 1.76; %W/(m^2.K)
```



## **Anexo B – Programação em ambiente gráfico**





```

function varargout = adicionarparedede(varargin)
% ADICIONARPAREDE MATLAB code for adicionarparedede.fig
%     ADICIONARPAREDE, by itself, creates a new ADICIONARPAREDE or raises the
existing
%     singleton*.
%
%     H = ADICIONARPAREDE returns the handle to a new ADICIONARPAREDE or the handle
to
%     the existing singleton*.
%
%     ADICIONARPAREDE('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in ADICIONARPAREDE.M with the given input arguments.
%
%     ADICIONARPAREDE('Property','Value',...) creates a new ADICIONARPAREDE or
raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before adicionarparedede_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to adicionarparedede_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help adicionarparedede

% Last Modified by GUIDE v2.5 21-Jul-2016 11:43:42

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @adicionarparedede_OpeningFcn, ...
                  'gui_OutputFcn',  @adicionarparedede_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before adicionarparedede is made visible.
function adicionarparedede_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to adicionarparedede (see VARARGIN)

% Choose default command line output for adicionarparedede
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

movegui(gcf, 'center')

set(handles.figure1, 'Name', 'Adicionar Parede, Teto, Cobertura e Pavimento')

set(handles.parede_tabela, 'Data', {})

% UIWAIT makes adicionarparede wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = adicionarparede_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function parede_nome_Callback(hObject, eventdata, handles)
% hObject handle to parede_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of parede_nome as text
% str2double(get(hObject, 'String')) returns contents of parede_nome as a
double

% --- Executes during object creation, after setting all properties.
function parede_nome_CreateFcn(hObject, eventdata, handles)
% hObject handle to parede_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiControlBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on selection change in parede_tipo_material.
function parede_tipo_material_Callback(hObject, eventdata, handles)
% hObject handle to parede_tipo_material (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String')) returns parede_tipo_material
contents as cell array
% contents{get(hObject, 'Value')} returns selected item from
parede_tipo_material

```

```

x1 = get(handles.parede_tipo_material, 'Value');

y1 = tipo_material(x1);

set(handles.parede_material, 'Value', 1);
set(handles.parede_material, 'String', y1);

clearvars x1 y1

% --- Executes during object creation, after setting all properties.
function parede_tipo_material_CreateFcn(hObject, eventdata, handles)
% hObject    handle to parede_tipo_material (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on selection change in parede_material.
function parede_material_Callback(hObject, eventdata, handles)
% hObject    handle to parede_material (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String')) returns parede_material contents✓
as cell array
%         contents{get(hObject, 'Value')} returns selected item from parede_material
x1 = get(handles.parede_tipo_material, 'Value');
x2 = get(handles.parede_material, 'Value');

[y2, y3] = rho(x1, x2);

set(handles.parede_rho, 'String', y2)
set(handles.parede_k, 'String', y3)

clearvars x1 x2 y2 y3

% --- Executes during object creation, after setting all properties.
function parede_material_CreateFcn(hObject, eventdata, handles)
% hObject    handle to parede_material (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function parede_rho_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to parede_rho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of parede_rho as text
%          str2double(get(hObject,'String')) returns contents of parede_rho as a double

% --- Executes during object creation, after setting all properties.
function parede_rho_CreateFcn(hObject, eventdata, handles)
% hObject      handle to parede_rho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function parede_k_Callback(hObject, eventdata, handles)
% hObject      handle to parede_k (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of parede_k as text
%          str2double(get(hObject,'String')) returns contents of parede_k as a double

% --- Executes during object creation, after setting all properties.
function parede_k_CreateFcn(hObject, eventdata, handles)
% hObject      handle to parede_k (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function parede_e_Callback(hObject, eventdata, handles)
% hObject      handle to parede_e (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of parede_e as text
%          str2double(get(hObject,'String')) returns contents of parede_e as a double

% --- Executes during object creation, after setting all properties.
function parede_e_CreateFcn(hObject, eventdata, handles)
% hObject      handle to parede_e (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in parede_adicionar_camada.
```

```
function parede_adicionar_camada_Callback(hObject, eventdata, handles)
% hObject handle to parede_adicionar_camada (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
x1 = get(handles.parede_tipo_material,'Value');
w2 = str2double(get(handles.parede_rho,'string'));
z2 = str2double(get(handles.parede_k,'string'));
q2 = str2double(get(handles.parede_e,'string'));
```

```
switch x1
    case 1
    case 2
        x4 = 'Água';
    case 3
        x4 = 'Argamassa';
    case 4
        x4 = 'Betão';
    case 5
        x4 = 'Borrachas';
    case 6
        x4 = 'Fibrocimento';
    case 7
        x4 = 'Gases';
    case 8
        x4 = 'Gessos';
    case 9
        x4 = 'Impermeabilização e Mastiques';
    case 10
        x4 = 'Inertes, solos e terras';
    case 11
        x4 = 'Isolamento Térmico';
    case 12
        x4 = 'Madeiras e Derivados';
    case 13
        x4 = 'Materiais Plásticos';
    case 14
        x4 = 'Material Cerâmico';
    case 15
        x4 = 'Metais';
    case 16
        x4 = 'Pedras';
    case 17
        x4 = 'Revestimentos de pisos ou paredes';
    case 18
        x4 = 'Vidro';
end
```

```
x2 = get(handles.parede_material,'value');
```

```
y4 = material_nome(x1,x2);
```

```
oldData = get(handles.parede_tabela, 'Data');  
newRow = {x4,y4,w2,z2,q2};  
newData = vertcat(oldData,newRow);  
set(handles.parede_tabela, 'Data',newData)
```

```
clearvars x1 x2 x4 w2 z2 q2 y4 oldData newRow newData
```

```
% --- Executes on button press in parede_remove_camada.  
function parede_remove_camada_Callback(hObject, eventdata, handles)  
% hObject    handle to parede_remove_camada (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

```
global tabledata  
tabledata = get(handles.parede_tabela, 'Data');
```

```
k = size(tabledata,1);
```

```
if k > 0  
tabledata(k,:) =[];  
else  
end
```

```
set(handles.parede_tabela, 'Data',tabledata)
```

```
clearvars k
```

```
% --- Executes on button press in parede_guardar.  
function parede_guardar_Callback(hObject, eventdata, handles)  
% hObject    handle to parede_guardar (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
load projecto.mat
```

```
global carproj
```

```
if get(handles.parede_check_uparede, 'Value')==1 % Introduzir valor do U
```

```
    switch get(handles.parede_tipo, 'Value')
```

```
        case 1
```

```
            warndlg('Selecione o tipo de elemento construtivo','Erro !')
```

```
        case 2 % Parede
```

```
            choice = questdlg('Deseja adicionar a parede?', 'Adicionar'  
Parede', 'Sim', 'Não', 'Cancelar', 'Cancelar');
```

```
            switch choice
```

```
                case 'Sim'
```

```
                    k = length(projecto(carproj).bdados.elementos)+1;
```

```
                    projecto(carproj).bdados.elementos(k).nome = get(handles.'
```

```
parede_nome, 'String');
```

```

        projecto(carproj).bdados.elementos(k).tipo = 'Parede';
        projecto(carproj).bdados.elementos(k).u = str2double(get(handles.parede_uvalor, 'String'));
        projecto(carproj).bdados.elementos(k).rsi = '-';
        projecto(carproj).bdados.elementos(k).rse = '-';
        projecto(carproj).bdados.elementos(k).tabela = [];

        save projecto.mat projecto
        warndlg('Parede adicionada', 'Aviso !')
        case 'Não'
        case 'Cancelar'
    end
    case 3 % Pavimento
        choice = questdlg('Deseja adicionar o pavimento?', 'Adicionar Pavimento', 'Sim', 'Não', 'Cancelar', 'Cancelar');

        switch choice
            case 'Sim'
                k = length(projecto(carproj).bdados.elementos)+1;
                projecto(carproj).bdados.elementos(k).nome = get(handles.parede_nome, 'String');
                projecto(carproj).bdados.elementos(k).tipo = 'Pavimento';
                projecto(carproj).bdados.elementos(k).u = str2double(get(handles.parede_uvalor, 'String'));
                projecto(carproj).bdados.elementos(k).rsi = '-';
                projecto(carproj).bdados.elementos(k).rse = '-';
                projecto(carproj).bdados.elementos(k).tabela = [];

                save projecto.mat projecto
                warndlg('Pavimento adicionado', 'Aviso !')
                case 'Não'
                case 'Cancelar'
            end
        case 4 % Cobertura
            choice = questdlg('Deseja adicionar a cobertura?', 'Adicionar Cobertura', 'Sim', 'Não', 'Cancelar', 'Cancelar');

            switch choice
                case 'Sim'
                    k = length(projecto(carproj).bdados.elementos)+1;
                    projecto(carproj).bdados.elementos(k).nome = get(handles.parede_nome, 'String');
                    projecto(carproj).bdados.elementos(k).tipo = 'Cobertura';
                    projecto(carproj).bdados.elementos(k).u = str2double(get(handles.parede_uvalor, 'String'));
                    projecto(carproj).bdados.elementos(k).rsi = '-';
                    projecto(carproj).bdados.elementos(k).rse = '-';
                    projecto(carproj).bdados.elementos(k).tabela = [];

                    save projecto.mat projecto
                    warndlg('Cobertura adicionada', 'Aviso !')
                    case 'Não'
                    case 'Cancelar'
                end
            case 5 % Tetos
                choice = questdlg('Deseja adicionar o teto?', 'Adicionar Cobertura', 'Sim', 'Não', 'Cancelar', 'Cancelar');

                switch choice
                    case 'Sim'

```

```

        k = length(projecto(carproj).bdados.elementos)+1;
        projecto(carproj).bdados.elementos(k).nome = get(handles.✓
parede_nome, 'String');
        projecto(carproj).bdados.elementos(k).tipo = 'Teto';
        projecto(carproj).bdados.elementos(k).u = str2double(get(✓
(handles.parede_uvalor, 'String'));
        projecto(carproj).bdados.elementos(k).rsi = '-';
        projecto(carproj).bdados.elementos(k).rse = '-';
        projecto(carproj).bdados.elementos(k).tabela = [];

        save projecto.mat projecto
        warndlg('Teto adicionado', 'Aviso !')
        case 'Não'
        case 'Cancelar'
    end
end

else % Adicionar elemento, mas fazendo o cálculo do U

    switch get(handles.parede_tipo, 'Value')
    case 1 % Selecione
        warndlg('Selecione o tipo de elemento construtivo', 'Erro !')
    case 2 % Parede
        choice = questdlg('Deseja adicionar a parede?', 'Adicionar✓
Parede', 'Sim', 'Não', 'Cancelar', 'Cancelar');

        switch choice
        case 'Sim'
            k = length(projecto(carproj).bdados.elementos)+1;
            projecto(carproj).bdados.elementos(k).nome = get(handles.✓
parede_nome, 'String');
            projecto(carproj).bdados.elementos(k).tipo = 'Parede';
            projecto(carproj).bdados.elementos(k).rsi = get(handles.✓
parede_rsi, 'String');
            projecto(carproj).bdados.elementos(k).rse = get(handles.✓
parede_rse, 'String');
            projecto(carproj).bdados.elementos(k).tabela = get(handles.✓
parede_tabela, 'Data');

            rsi = str2double(get(handles.parede_rsi, 'String'));
            rse = str2double(get(handles.parede_rse, 'String'));
            tabela = get(handles.parede_tabela, 'Data');

            u=u_elementos(rsi, rse, tabela);

            projecto(carproj).bdados.elementos(k).u=u;

            save projecto.mat projecto
            warndlg('Parede adicionada', 'Aviso !')
            case 'Não'
            case 'Cancelar'
        end
    case 3 % Pavimento
        choice = questdlg('Deseja adicionar o pavimento?', 'Adicionar✓
Parede', 'Sim', 'Não', 'Cancelar', 'Cancelar');

        switch choice
        case 'Sim'

```



```

        k = length(projecto(carproj).bdados.elementos)+1;
        projecto(carproj).bdados.elementos(k).nome = get(handles.✓
parede_nome, 'String');
        projecto(carproj).bdados.elementos(k).tipo = 'Pavimento';
        projecto(carproj).bdados.elementos(k).rsi = get(handles.✓
parede_rsi, 'String');
        projecto(carproj).bdados.elementos(k).rse = get(handles.✓
parede_rse, 'String');
        projecto(carproj).bdados.elementos(k).tabela = get(handles.✓
parede_tabela, 'Data');

        rsi = str2double(get(handles.parede_rsi, 'String'));
        rse = str2double(get(handles.parede_rse, 'String'));
        tabela = get(handles.parede_tabela, 'Data');

        u=u_elementos(rsi,rse,tabela);

        projecto(carproj).bdados.elementos(k).u=u;

        save projecto.mat projecto
        warndlg('Pavimento adicionado', 'Aviso !')
        case 'Não'
        case 'Cancelar'
    end
    case 4 % Cobertura
        choice = questdlg('Deseja adicionar a cobertura?', 'Adicionar✓
Parede', 'Sim', 'Não', 'Cancelar', 'Cancelar');

        switch choice
            case 'Sim'
                k = length(projecto(carproj).bdados.elementos)+1;
                projecto(carproj).bdados.elementos(k).nome = get(handles.✓
parede_nome, 'String');
                projecto(carproj).bdados.elementos(k).tipo = 'Cobertura';
                projecto(carproj).bdados.elementos(k).rsi = get(handles.✓
parede_rsi, 'String');
                projecto(carproj).bdados.elementos(k).rse = get(handles.✓
parede_rse, 'String');
                projecto(carproj).bdados.elementos(k).tabela = get(handles.✓
parede_tabela, 'Data');

                rsi = str2double(get(handles.parede_rsi, 'String'));
                rse = str2double(get(handles.parede_rse, 'String'));
                tabela = get(handles.parede_tabela, 'Data');

                u=u_elementos(rsi,rse,tabela);

                projecto(carproj).bdados.elementos(k).u=u;

                save projecto.mat projecto
                warndlg('Cobertura adicionada', 'Aviso !')
                case 'Não'
                case 'Cancelar'
            end

            case 5 %Teto
                choice = questdlg('Deseja adicionar o teto?', 'Adicionar✓
Parede', 'Sim', 'Não', 'Cancelar', 'Cancelar');

                switch choice

```

```

        case 'Sim'
            k = length(projecto(carproj).bdados.elementos)+1;
            projecto(carproj).bdados.elementos(k).nome = get(handles.✓
parede_nome, 'String');
            projecto(carproj).bdados.elementos(k).tipo = 'Teto';
            projecto(carproj).bdados.elementos(k).rsi = get(handles.✓
parede_rsi, 'String');
            projecto(carproj).bdados.elementos(k).rse = get(handles.✓
parede_rse, 'String');
            projecto(carproj).bdados.elementos(k).tabela = get(handles.✓
parede_tabela, 'Data');

            rsi = str2double(get(handles.parede_rsi, 'String'));
            rse = str2double(get(handles.parede_rse, 'String'));
            tabela = get(handles.parede_tabela, 'Data');

            u=u_elementos(rsi,rse,tabela);

            projecto(carproj).bdados.elementos(k).u=u;

            save projecto.mat projecto
            warndlg('Teto adicionado', 'Aviso !')
        case 'Não'
        case 'Cancelar'
    end
end
end

clearvars choice k rsi rse tabela u

```

```

function parede_rsi_Callback(hObject, eventdata, handles)
% hObject      handle to parede_rsi (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of parede_rsi as text
%         str2double(get(hObject,'String')) returns contents of parede_rsi as a ✓
double

```

```

% --- Executes during object creation, after setting all properties.
function parede_rsi_CreateFcn(hObject, eventdata, handles)
% hObject      handle to parede_rsi (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function parede_rse_Callback(hObject, eventdata, handles)
% hObject      handle to parede_rse (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of parede_rse as text
%         str2double(get(hObject,'String')) returns contents of parede_rse as a
double

% --- Executes during object creation, after setting all properties.
function parede_rse_CreateFcn(hObject, eventdata, handles)
% hObject      handle to parede_rse (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in parede_tipo.
function parede_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to parede_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns parede_tipo contents as
cell array
%         contents{get(hObject,'Value')} returns selected item from parede_tipo

switch get(handles.parede_tipo,'Value')
    case 1
        set(handles.parede_guardar,'String','Guardar')
    case 2
        set(handles.parede_guardar,'String','Guardar Parede')
    case 3
        set(handles.parede_guardar,'String','Guardar Pavimento')
    case 4
        set(handles.parede_guardar,'String','Guardar Cobertura')
    case 5
        set(handles.parede_guardar,'String','Guardar Teto')
end

% --- Executes during object creation, after setting all properties.
function parede_tipo_CreateFcn(hObject, eventdata, handles)
% hObject      handle to parede_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----

```

```

function adparede_inicio_Callback(hObject, eventdata, handles)
% hObject      handle to adparede_inicio (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('inicio.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_local_Callback(hObject, eventdata, handles)
% hObject      handle to adparede_local (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('local.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_materiais_Callback(hObject, eventdata, handles)
% hObject      handle to adparede_materiais (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('registamateriais.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_relatorios_Callback(hObject, eventdata, handles)
% hObject      handle to adparede_relatorios (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to adparede_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('vidro.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_caixilho_Callback(hObject, eventdata, handles)
% hObject      handle to adparede_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('caixilho.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

```

```

% -----
function adparede_adporta_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_adporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('Porta.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_verporta_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_verporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('porta_vereliminae.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_simples_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_simples (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_simples.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_dupla_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_dupla (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_dupla.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_duplafolha_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_duplafolha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_verjanela_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_verjanela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

```

```

% -----
function adparede_adhorario_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_adhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_horario.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function aadparede_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to aadparede_defhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_defespaco_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_defespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_espaco.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% -----
function adparede_verparede_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_verparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_parede.m')

close(findobj('type','figure','name','Adicionar Parede, Teto, Cobertura e Pavimento'))

% --- Executes on button press in parede_check_uparede.
function parede_check_uparede_Callback(hObject, eventdata, handles)
% hObject    handle to parede_check_uparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hint: get(hObject,'Value') returns toggle state of parede_check_uparede

if get(handles.parede_check_uparede,'Value')==1
    set(handles.parede_uvalor,'Visible','On')
    set(handles.parede_text13,'Visible','On')
else
    set(handles.parede_uvalor,'Visible','Off')
    set(handles.parede_text13,'Visible','Off')
end

function parede_uvalor_Callback(hObject, eventdata, handles)
% hObject    handle to parede_uvalor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of parede_uvalor as text
%         str2double(get(hObject,'String')) returns contents of parede_uvalor as a double

% --- Executes during object creation, after setting all properties.
function parede_uvalor_CreateFcn(hObject, eventdata, handles)
% hObject    handle to parede_uvalor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function adparede_elementos_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_elementos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function adparede_espaco_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_espaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function adparede_horario_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_horario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function adparede_janela_Callback(hObject, eventdata, handles)
% hObject    handle to adparede_janela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```
% -----  
function adparede_porta_Callback(hObject, eventdata, handles)  
% hObject    handle to adparede_porta (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% -----  
function adparede_parede_Callback(hObject, eventdata, handles)  
% hObject    handle to adparede_parede (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% -----  
function adparede_Callback(hObject, eventdata, handles)  
% hObject    handle to adparede (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```



```

function varargout = caixilho(varargin)
% CAIXILHO MATLAB code for caixilho.fig
%     CAIXILHO, by itself, creates a new CAIXILHO or raises the existing
%     singleton*.
%
%     H = CAIXILHO returns the handle to a new CAIXILHO or the handle to
%     the existing singleton*.
%
%     CAIXILHO('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in CAIXILHO.M with the given input arguments.
%
%     CAIXILHO('Property','Value',...) creates a new CAIXILHO or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before caixilho_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to caixilho_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help caixilho

% Last Modified by GUIDE v2.5 21-Jul-2016 05:18:21

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @caixilho_OpeningFcn, ...
                  'gui_OutputFcn',  @caixilho_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before caixilho is made visible.
function caixilho_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to caixilho (see VARARGIN)

% Choose default command line output for caixilho
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```

```

movegui(gcf,'center')

set(handles.figure1,'Name','Caixilho')

set(handles.caixilho_panel_ver,'visible','off')
set(handles.caixilho_panel_adicionar,'visible','off')

% UIWAIT makes caixilho wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = caixilho_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in caixilho_adicionar.
function caixilho_adicionar_Callback(hObject, eventdata, handles)
% hObject handle to caixilho_adicionar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
set(handles.figure1,'Name','Adicionar Caixilho')

set(handles.caixilho_panel_adicionar,'Visible','On')
set(handles.caixilho_panel_ver,'Visible','Off')
set(handles.caixilho_text8,'Visible','On')
set(handles.caixilho_nome,'Visible','On')

% --- Executes on button press in caixilho_ver.
function caixilho_ver_Callback(hObject, eventdata, handles)
% hObject handle to caixilho_ver (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

set(handles.figure1,'Name','Ver / Eliminar Caixilho')

set(handles.caixilho_panel_adicionar,'Visible','Off')
set(handles.caixilho_panel_ver,'Visible','On')
set(handles.caixilho_text8,'Visible','Off')
set(handles.caixilho_nome,'Visible','Off')

load projecto.mat
global carproj

set(handles.caixilho_lista,'String',{projecto(carproj).bdados.caixilho.nome})

% --- Executes on button press in caixilho_guardar.
function caixilho_guardar_Callback(hObject, eventdata, handles)
% hObject handle to caixilho_guardar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

choice = questdlg('Tem a certeza que deseja adicionar o ✓
caixilho','','Sim','Não','Cancelar','Cancelar');

switch choice

    case 'Sim'

load projecto.mat

global carproj
% Introduzir o valor de U da caixilharia

if get(handles.caixilho_check_uf,'Value')==1

    k = length(projecto(carproj).bdados.caixilho)+1;

    projecto(carproj).bdados.caixilho(k).nome = get(handles.caixilho_nome,'String');
    projecto(carproj).bdados.caixilho(k).uf = get(handles.caixilho_uf,'String');

elseif get(handles.caixilho_eqgeral,'Value')==1 %Caixilharia geral
    x1=str2double(get(handles.caixilho_ubigeral,'String'));
    x2=str2double(get(handles.caixilho_upgeral,'String'));
    x3=str2double(get(handles.caixilho_bfgeral,'String'));
    x4=str2double(get(handles.caixilho_bpgeral,'String'));

    uf = ((x1)-(x2*x4))/x3;

    k = length(projecto(carproj).bdados.caixilho)+1;

    projecto(carproj).bdados.caixilho(k).nome = get(handles.caixilho_nome,'String');
    projecto(carproj).bdados.caixilho(k).uf = uf;
end

save projecto.mat projecto

warndlg('O caixilho foi adicionado','Aviso!')

    case 'Não'
    case 'Cancelar'
end

clearvars uf x1 x2 x3 x4 y1 y2 y3 y4 k choice

% --- Executes on button press in caixilho_check_uf.
function caixilho_check_uf_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_check_uf (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
set(handles.caixilho_eqgeral,'Value',0)

set(handles.caixilho_panelnormal,'Visible','Off')

if get(handles.caixilho_check_uf,'Value')==1
    set(handles.caixilho_uf,'Visible','On')
    set(handles.caixilho_text18,'Visible','On')

```

```

else
    set(handles.caixilho_uf, 'Visible', 'Off')
    set(handles.caixilho_text18, 'Visible', 'Off')
end
% Hint: get(hObject, 'Value') returns toggle state of caixilho_check_uf

function caixilho_uf_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_uf (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of caixilho_uf as text
%          str2double(get(hObject, 'String')) returns contents of caixilho_uf as a
double

% --- Executes during object creation, after setting all properties.
function caixilho_uf_CreateFcn(hObject, eventdata, handles)
% hObject      handle to caixilho_uf (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function caixilho_nome_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_nome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of caixilho_nome as text
%          str2double(get(hObject, 'String')) returns contents of caixilho_nome as a
double

% --- Executes during object creation, after setting all properties.
function caixilho_nome_CreateFcn(hObject, eventdata, handles)
% hObject      handle to caixilho_nome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on selection change in caixilho_lista.
function caixilho_lista_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_lista (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns caixilho_lista contents✓
as cell array
%           contents{get(hObject,'Value')} returns selected item from caixilho_lista

% --- Executes during object creation, after setting all properties.
function caixilho_lista_CreateFcn(hObject, eventdata, handles)
% hObject      handle to caixilho_lista (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in caixilho_verpush.
function caixilho_verpush_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_verpush (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

load projecto.mat

global carproj

k = get(handles.caixilho_lista,'Value');

set(handles.caixilho_vernome,'String',projecto(carproj).bdados.caixilho(k).nome)
set(handles.caixilho_veruf,'String',projecto(carproj).bdados.caixilho(k).uf)

clearvars k

% --- Executes on button press in caixilho_eliminar.
function caixilho_eliminar_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_eliminar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

if get(handles.caixilho_lista,'Value')>5
    choice = questdlg('Deseja eliminar o✓
caixilho','', 'Sim', 'Não', 'Cancelar', 'Cancelar');

    switch choice
        case 'Sim'
            load projecto.mat

            global carproj

            k = get(handles.caixilho_lista,'Value');

            projecto(carproj).bdados.caixilho(k)=[];

```

```

        save projecto.mat projecto

        set(handles.caixilho_lista, 'Value', 1)

        set(handles.caixilho_lista, 'String', {projecto(carproj).bdados.caixilho.✓
nome});

        warndlg('Caixilho foi eliminado!', 'Aviso!')
            case 'Não'
            case 'Cancelar'
        end
    else
        errordlg('Não é possível eliminar a caixilharia', 'Aviso!')
    end

clearvars choice k

function caixilho_vernome_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_vernome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of caixilho_vernome as text
%          str2double(get(hObject, 'String')) returns contents of caixilho_vernome as a ✓
double

% --- Executes during object creation, after setting all properties.
function caixilho_vernome_CreateFcn(hObject, eventdata, handles)
% hObject      handle to caixilho_vernome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get✓
(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function caixilho_veruf_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_veruf (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of caixilho_veruf as text
%          str2double(get(hObject, 'String')) returns contents of caixilho_veruf as a ✓
double

% --- Executes during object creation, after setting all properties.
function caixilho_veruf_CreateFcn(hObject, eventdata, handles)
% hObject      handle to caixilho_veruf (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

```

```

%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function caixilho_inicio_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_inicio (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('inicio.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))

% -----
function caixilho_local_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_local (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('local.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))
% -----
function caixilho_materiais_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_materiais (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('registamateriais.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))

% -----
function caixilho_relatorios_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_relatorios (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('relatorios.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))

% -----
function caixilho_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

run('vidro.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))

% -----
function caixilho_caixilho_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function caixilho_adporta_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_adporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('Porta.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))
% -----
function caixilho_verporta_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_verporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('porta_vereliminar.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))

% -----
function caixilho_jansimples_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_jansimples (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_simples.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))

% -----
function caixilho_jandupla_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_jandupla (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('janela_dupla.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))

% -----
function caixilho_duplafolha_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_duplafolha (see GCBO)

```



```
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
run('janela_duplafolhamesmoaro.m')
```

```
set(handles.figure1,'Name','Caixilho')
```

```
close(findobj('type','figure','name','Caixilho'))
```

```
% -----
function caixilho_verjanela_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_verjanela (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')
```

```
set(handles.figure1,'Name','Caixilho')
```

```
close(findobj('type','figure','name','Caixilho'))
```

```
% -----
function caixilho_adhorario_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_adhorario (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('definir_horario.m')
```

```
set(handles.figure1,'Name','Caixilho')
```

```
close(findobj('type','figure','name','Caixilho'))
```

```
% -----
function caixilho_defhorario_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_defhorario (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('horario.m')
```

```
set(handles.figure1,'Name','Caixilho')
```

```
close(findobj('type','figure','name','Caixilho'))
```

```
% -----
function caixilho_defespaco_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_defespaco (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('definir_espaco.m')
```

```
set(handles.figure1,'Name','Caixilho')
```

```
close(findobj('type','figure','name','Caixilho'))
```

```
% -----
function caixilho_verespaco_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_verespaco (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('ver_espaco.m')
```

```

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))

% -----
function caixilho_adprede_Callback(hObject, eventdata, handles)
% hObject    handle to caixilho_adprede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))

% -----
function caixilho_verparede_Callback(hObject, eventdata, handles)
% hObject    handle to caixilho_verparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_parede.m')

set(handles.figure1,'Name','Caixilho')

close(findobj('type','figure','name','Caixilho'))

% --- Executes on button press in caixilho_eqgeral.
function caixilho_eqgeral_Callback(hObject, eventdata, handles)
% hObject    handle to caixilho_eqgeral (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of caixilho_eqgeral

set(handles.caixilho_check_uf,'Value',0)

if get(handles.caixilho_eqgeral,'Value')==1

set(handles.caixilho_uf,'Visible','Off')
set(handles.caixilho_text18,'Visible','Off')

set(handles.caixilho_panelnormal,'Visible','On')
else

set(handles.caixilho_panelnormal,'Visible','Off')

end

function caixilho_ubigeral_Callback(hObject, eventdata, handles)
% hObject    handle to caixilho_ubigeral (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of caixilho_ubigeral as text
%        str2double(get(hObject,'String')) returns contents of caixilho_ubigeral as a double

```

```

% --- Executes during object creation, after setting all properties.
function caixilho_ubigeral_CreateFcn(hObject, eventdata, handles)
% hObject    handle to caixilho_ubigeral (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function caixilho_upgeral_Callback(hObject, eventdata, handles)
% hObject    handle to caixilho_upgeral (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of caixilho_upgeral as text
%         str2double(get(hObject,'String')) returns contents of caixilho_upgeral as a
double

% --- Executes during object creation, after setting all properties.
function caixilho_upgeral_CreateFcn(hObject, eventdata, handles)
% hObject    handle to caixilho_upgeral (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function caixilho_bfgeral_Callback(hObject, eventdata, handles)
% hObject    handle to caixilho_bfgeral (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of caixilho_bfgeral as text
%         str2double(get(hObject,'String')) returns contents of caixilho_bfgeral as a
double

% --- Executes during object creation, after setting all properties.
function caixilho_bfgeral_CreateFcn(hObject, eventdata, handles)
% hObject    handle to caixilho_bfgeral (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function caixilho_bpgeral_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_bpgeral (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of caixilho_bpgeral as text
%          str2double(get(hObject,'String')) returns contents of caixilho_bpgeral as a
double

% --- Executes during object creation, after setting all properties.
function caixilho_bpgeral_CreateFcn(hObject, eventdata, handles)
% hObject      handle to caixilho_bpgeral (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function caixilho_text20_CreateFcn(hObject, eventdata, handles)
% hObject      handle to caixilho_text20 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% -----
function caixilho_elementos_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_elementos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function caixilho_espaco_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_espaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function caixilho_horario_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_horario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----

```

```
function caixilho_janela_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_janela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function caixilho_porta_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_porta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function caixilho_parede_Callback(hObject, eventdata, handles)
% hObject      handle to caixilho_parede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```

function varargout = definir_espaco(varargin)
% DEFINIR_ESPACO MATLAB code for definir_espaco.fig
%     DEFINIR_ESPACO, by itself, creates a new DEFINIR_ESPACO or raises the
existing
%     singleton*.
%
%     H = DEFINIR_ESPACO returns the handle to a new DEFINIR_ESPACO or the handle
to
%     the existing singleton*.
%
%     DEFINIR_ESPACO('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in DEFINIR_ESPACO.M with the given input arguments.
%
%     DEFINIR_ESPACO('Property','Value',...) creates a new DEFINIR_ESPACO or raises
the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before definir_espaco_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to definir_espaco_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help definir_espaco

% Last Modified by GUIDE v2.5 21-Jul-2016 11:18:05

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @definir_espaco_OpeningFcn, ...
                  'gui_OutputFcn',  @definir_espaco_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before definir_espaco is made visible.
function definir_espaco_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to definir_espaco (see VARARGIN)

% Choose default command line output for definir_espaco
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes definir_espaco wait for user response (see UIRESUME)
% uiwait(handles.figure1);

movegui(gcf, 'center')

set(handles.figure1, 'Name', 'Adicionar Espaço')

load projecto.mat

global carproj

projecto(carproj).elementos.parede=[];
projecto(carproj).elementos.pavimento=[];
projecto(carproj).elementos.cobertura=[];
projecto(carproj).elementos.teto=[];

s1 = 'Parede';
s2 = 'Pavimento';
s3 = 'Cobertura';
s4 = 'Teto';

k = length(projecto(carproj).bdados.elementos);

for x=1:k

    if strcmp(projecto(carproj).bdados.elementos(x).tipo,s1) ==1

        z = length(projecto(carproj).elementos.parede)+1;

        projecto(carproj).elementos.parede(z).nome = projecto(carproj).bdados.✓
elementos(x).nome;
        projecto(carproj).elementos.parede(z).tipo = projecto(carproj).bdados.✓
elementos(x).tipo;
        projecto(carproj).elementos.parede(z).u = projecto(carproj).bdados.elementos✓
(x).u;

        elseif strcmp(projecto(carproj).bdados.elementos(x).tipo,s2) ==1

            z = length(projecto(carproj).elementos.pavimento)+1;

            projecto(carproj).elementos.pavimento(z).nome = projecto(carproj).bdados.✓
elementos(x).nome;
            projecto(carproj).elementos.pavimento(z).tipo = projecto(carproj).bdados.✓
elementos(x).tipo;
            projecto(carproj).elementos.pavimento(z).u = projecto(carproj).bdados.✓
elementos(x).u;

            elseif strcmp(projecto(carproj).bdados.elementos(x).tipo,s3) ==1

                z = length(projecto(carproj).elementos.cobertura)+1;

                projecto(carproj).elementos.cobertura(z).nome = projecto(carproj).bdados.✓
elementos(x).nome;
                projecto(carproj).elementos.cobertura(z).tipo = projecto(carproj).bdados.✓
elementos(x).tipo;

```

```

        proyecto(carproj).elementos.cobertura(z).u = proyecto(carproj).bdados.✓
elementos(x).u;

elseif strcmp(proyecto(carproj).bdados.elementos(x).tipo,s4) ==1

    z = length(proyecto(carproj).elementos.teto)+1;

    proyecto(carproj).elementos.teto(z).nome = proyecto(carproj).bdados.✓
elementos(x).nome;
    proyecto(carproj).elementos.teto(z).tipo = proyecto(carproj).bdados.✓
elementos(x).tipo;
    proyecto(carproj).elementos.teto(z).u = proyecto(carproj).bdados.elementos✓
(x).u;
    end
end

save proyecto.mat proyecto

clearvars s1 s2 s3 s4 k z x

% --- Outputs from this function are returned to the command line.
function varargout = definir_espaco_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function defespaco_nome_Callback(hObject, eventdata, handles)
% hObject handle to defespaco_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_nome as text
% str2double(get(hObject,'String')) returns contents of defespaco_nome as a✓
double

% --- Executes during object creation, after setting all properties.
function defespaco_nome_CreateFcn(hObject, eventdata, handles)
% hObject handle to defespaco_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function adespaco_inicio_Callback(hObject, eventdata, handles)
% hObject handle to adespaco_inicio (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB

```



```

% handles      structure with handles and user data (see GUIDATA)
run('inicio.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_local_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_local (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('local.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_materiais_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_materiais (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('registamateriais.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function defespaco_relatorios_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_relatorios (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('vidro.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_caixilho_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('caixilho.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_adporta_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_adporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('Porta.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----

```

```

function adespaco_verporta_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_verporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('porta_vereliminar.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_simples_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_simples (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_simples.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_dupla_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_dupla (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_dupla.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_duplafolha_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_duplafolha (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_verjanela_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_verjanela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% --- Executes on button press in defespaco_exelement1_check.
function defespaco_exelement1_check_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement1_check (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_exelement1_check

% --- Executes on button press in defespaco_exelement2_check.
function defespaco_exelement2_check_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement2_check (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_exelement2_check

```

```

% --- Executes on selection change in defespaco_exelement1_tipo.
function defespaco_exelement1_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement1_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_exelement1_tipo
contents as cell array
%      contents{get(hObject,'Value')} returns selected item from
defespaco_exelement1_tipo
load projecto.mat
global carproj

if get(handles.defespaco_exelement1_check,'Value')==1

    switch get(handles.defespaco_exelement1_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_exelement1_elemento,'Value',1)
            set(handles.defespaco_exelement1_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_exelement1_elemento,'Value',1)
                set(handles.defespaco_exelement1_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement1_elemento,'Value',1)
                set(handles.defespaco_exelement1_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.parede.nome})
            end
        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_exelement1_elemento,'Value',1)
                set(handles.defespaco_exelement1_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement1_elemento,'Value',1)
                set(handles.defespaco_exelement1_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.pavimento.nome})
            end
        case 4 %Cobertura

            if isempty(projecto(carproj).elementos.cobertura)==1
                set(handles.defespaco_exelement1_elemento,'Value',1)
                set(handles.defespaco_exelement1_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement1_elemento,'Value',1)
                set(handles.defespaco_exelement1_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.cobertura.nome})
            end
        case 5 %Teto

            if isempty(projecto(carproj).elementos.teto)==1
                set(handles.defespaco_exelement1_elemento,'Value',1)
                set(handles.defespaco_exelement1_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement1_elemento,'Value',1)
                set(handles.defespaco_exelement1_elemento,'String',{'(Selezione)',

```

```

projecto(carproj).elementos.teto.nome})
    end
    case 6 %Janela
        set(handles.defespaco_exelement1_elemento, 'Value', 1)
        set(handles.defespaco_exelement1_elemento, 'String', {'(Selecione)', '↙
projecto(carproj).bdados.janela.nome})
    case 7 %Porta
        set(handles.defespaco_exelement1_elemento, 'Value', 1)
        set(handles.defespaco_exelement1_elemento, 'String', {'(Selecione)', '↙
projecto(carproj).bdados.porta.nome})
    end
end

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement1_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement1_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%      See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

% --- Executes on selection change in defespaco_exelement1_elemento.
function defespaco_exelement1_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement1_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject, 'String')) returns
defespaco_exelement1_elemento contents as cell array
%      contents{get(hObject, 'Value')} returns selected item from
defespaco_exelement1_elemento

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement1_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement1_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%      See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

% --- Executes on button press in defespaco_exelement3_check.
function defespaco_exelement3_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement3_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hint: get(hObject, 'Value') returns toggle state of defespaco_exelement3_check

```

```

% --- Executes on button press in defespaco_exelement4_check.
function defespaco_exelement4_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement4_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_exelement4_check

% --- Executes on button press in defespaco_exelement5_check.
function defespaco_exelement5_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement5_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_exelement5_check

% --- Executes on button press in defespaco_exelement6_check.
function defespaco_exelement6_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement6_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_exelement6_check

% --- Executes on button press in defespaco_exelement7_check.
function defespaco_exelement7_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement7_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_exelement7_check

% --- Executes on button press in defespaco_exelement8_check.
function defespaco_exelement8_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement8_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_exelement8_check

function defespaco_exelement1_quantidade_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement1_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement1_quantidade
as text
%         str2double(get(hObject,'String')) returns contents of

```

```
defespaco_exelement1_quantidade as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function defespaco_exelement1_quantidade_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to defespaco_exelement1_quantidade (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
```

```
(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
function defespaco_exelement1_area_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to defespaco_exelement1_area (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of defespaco_exelement1_area as text
```

```
%         str2double(get(hObject,'String')) returns contents of
```

```
defespaco_exelement1_area as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function defespaco_exelement1_area_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to defespaco_exelement1_area (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
```

```
(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
function defespaco_numocupacao_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to defespaco_numocupacao (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of defespaco_numocupacao as text
```

```
%         str2double(get(hObject,'String')) returns contents of defespaco_numocupacao
```

```
as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function defespaco_numocupacao_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to defespaco_numocupacao (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```

%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_ocupunid.
function defespaco_ocupunid_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_ocupunid (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_ocupunid
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
defespaco_ocupunid

% --- Executes during object creation, after setting all properties.
function defespaco_ocupunid_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_ocupunid (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_ocupsensivel_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_ocupsensivel (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_ocupsensivel as text
%         str2double(get(hObject,'String')) returns contents of
defespaco_ocupsensivel as a double

% --- Executes during object creation, after setting all properties.
function defespaco_ocupsensivel_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_ocupsensivel (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_ocuplatente_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to defespaco_ocuplatente (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_ocuplatente as text
%          str2double(get(hObject,'String')) returns contents of defespaco_ocuplatente
as a double

% --- Executes during object creation, after setting all properties.
function defespaco_ocuplatente_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_ocuplatente (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_area_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_area as text
%          str2double(get(hObject,'String')) returns contents of defespaco_area as a
double

% --- Executes during object creation, after setting all properties.
function defespaco_area_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_elepotunid.
function defespaco_elepotunid_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_elepotunid (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_elepotunid
contents as cell array
%          contents{get(hObject,'Value')} returns selected item from
defespaco_elepotunid

```



```

% --- Executes during object creation, after setting all properties.
function defespaco_elepotunid_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_elepotunid (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_elepot_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_elepot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_elepot as text
%         str2double(get(hObject,'String')) returns contents of defespaco_elepot as a
double

% --- Executes during object creation, after setting all properties.
function defespaco_elepot_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_elepot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_disensivel_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_disensivel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_disensivel as text
%         str2double(get(hObject,'String')) returns contents of defespaco_disensivel
as a double

% --- Executes during object creation, after setting all properties.
function defespaco_disensivel_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_disensivel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))

```

```

    set(hObject,'BackgroundColor','white');
end

```

```

function defespaco_dilatente_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_dilatente (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_dilatente as text
%          str2double(get(hObject,'String')) returns contents of defespaco_dilatente
as a double

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_dilatente_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_dilatente (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in defespaco_ilumunid.

```

```

function defespaco_ilumunid_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_ilumunid (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_ilumunid
contents as cell array
%          contents{get(hObject,'Value')} returns selected item from
defespaco_ilumunid

```

```

% --- Executes during object creation, after setting all properties.

```

```

function defespaco_ilumunid_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_ilumunid (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function defespaco_ilupot_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_ilupot (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of defespaco_ilupot as text
%         str2double(get(hObject,'String')) returns contents of defespaco_ilupot as a
double

% --- Executes during object creation, after setting all properties.
function defespaco_ilupot_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_ilupot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in defespaco_intelement1_check.
function defespaco_intelement1_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement1_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_intelement1_check

% --- Executes on button press in defespaco_intelement2_check.
function defespaco_intelement2_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement2_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_intelement2_check

% --- Executes on selection change in defespaco_intelement1_tipo.
function defespaco_intelement1_tipo_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement1_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
defespaco_intelement1_tipo contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
defespaco_intelement1_tipo
load projecto.mat
global carproj

if get(handles.defespaco_intelement1_check,'Value')==1

    switch get(handles.defespaco_intelement1_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_intelement1_elemento,'Value',1)
            set(handles.defespaco_intelement1_elemento,'String','(Selezione)')
        case 2 %Parede

```

```

        if isempty(projecto(carproj).elementos.paredes)==1
            set(handles.defespaco_intelement1_elemento,'Value',1)
            set(handles.defespaco_intelement1_elemento,'String','(Selecione)')
        else
            set(handles.defespaco_intelement1_elemento,'Value',1)
            set(handles.defespaco_intelement1_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.paredes.nome})
        end

        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_intelement1_elemento,'Value',1)
                set(handles.defespaco_intelement1_elemento,'String','(Selecione)')
            else
                set(handles.defespaco_intelement1_elemento,'Value',1)
                set(handles.defespaco_intelement1_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.pavimento.nome})
            end

            case 4 %Cobertura

                if isempty(projecto(carproj).elementos.cobertura)==1
                    set(handles.defespaco_intelement1_elemento,'Value',1)
                    set(handles.defespaco_intelement1_elemento,'String','(Selecione)')
                else
                    set(handles.defespaco_intelement1_elemento,'Value',1)
                    set(handles.defespaco_intelement1_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.cobertura.nome})
                end

                case 5 %Teto

                    if isempty(projecto(carproj).elementos.teto)==1
                        set(handles.defespaco_intelement1_elemento,'Value',1)
                        set(handles.defespaco_intelement1_elemento,'String','(Selecione)')
                    else
                        set(handles.defespaco_intelement1_elemento,'Value',1)
                        set(handles.defespaco_intelement1_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.teto.nome})
                    end

                    case 6 %Janela

                        set(handles.defespaco_intelement1_elemento,'Value',1)
                        set(handles.defespaco_intelement1_elemento,'String',{'(Selecione)',↵
projecto(carproj).bdados.janela.nome})
                    case 7 %Porta

                        set(handles.defespaco_intelement1_elemento,'Value',1)
                        set(handles.defespaco_intelement1_elemento,'String',{'(Selecione)',↵
projecto(carproj).bdados.porta.nome})
                    end
                end
            end
        end
    end
end

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement1_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement1_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_intelement1_elemento.
function defespaco_intelement1_elemento_Callback(hObject, eventdata, handles)
% hObject     handle to defespaco_intelement1_elemento (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
defespaco_intelement1_elemento contents as cell array
%     contents{get(hObject,'Value')} returns selected item from
defespaco_intelement1_elemento

% --- Executes during object creation, after setting all properties.
function defespaco_intelement1_elemento_CreateFcn(hObject, eventdata, handles)
% hObject     handle to defespaco_intelement1_elemento (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in defespaco_intelement3_check.
function defespaco_intelement3_check_Callback(hObject, eventdata, handles)
% hObject     handle to defespaco_intelement3_check (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_intelement3_check

% --- Executes on button press in defespaco_intelement4_check.
function defespaco_intelement4_check_Callback(hObject, eventdata, handles)
% hObject     handle to defespaco_intelement4_check (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_intelement4_check

% --- Executes on button press in defespaco_intelement5_check.
function defespaco_intelement5_check_Callback(hObject, eventdata, handles)
% hObject     handle to defespaco_intelement5_check (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_intelement5_check

```

```

% --- Executes on button press in defespaco_intelement6_check.
function defespaco_intelement6_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement6_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_intelement6_check

% --- Executes on button press in defespaco_intelement7_check.
function defespaco_intelement7_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement7_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_intelement7_check

% --- Executes on button press in defespaco_intelement8_check.
function defespaco_intelement8_check_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement8_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of defespaco_intelement8_check

function defespaco_intelement1_quantidade_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement1_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement1_quantidade
as text
%         str2double(get(hObject,'String')) returns contents of
defespaco_intelement1_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement1_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement1_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement1_area_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement1_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of defespaco_intelement1_area as✓
text
%         str2double(get(hObject,'String')) returns contents of✓
defespaco_intelement1_area as a double


% --- Executes during object creation, after setting all properties.
function defespaco_intelement1_area_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement1_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function defespaco_intelement1_temp_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement1_temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of defespaco_intelement1_temp as✓
text
%         str2double(get(hObject,'String')) returns contents of✓
defespaco_intelement1_temp as a double


% --- Executes during object creation, after setting all properties.
function defespaco_intelement1_temp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement1_temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function defespaco_ti_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_ti (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of defespaco_ti as text
%         str2double(get(hObject,'String')) returns contents of defespaco_ti as a✓
double


% --- Executes during object creation, after setting all properties.
function defespaco_ti_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_ti (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_hi_Callback(hObject, eventdata, handles)
% hObject handle to defespaco_hi (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_hi as text
% str2double(get(hObject,'String')) returns contents of defespaco_hi as a
double

% --- Executes during object creation, after setting all properties.
function defespaco_hi_CreateFcn(hObject, eventdata, handles)
% hObject handle to defespaco_hi (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_exelement2_tipo.
function defespaco_exelement2_tipo_Callback(hObject, eventdata, handles)
% hObject handle to defespaco_exelement2_tipo (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_exelement2_tipo
contents as cell array
% contents{get(hObject,'Value')} returns selected item from
defespaco_exelement2_tipo
load projecto.mat
global carproj

if get(handles.defespaco_exelement2_check,'Value')==1

    switch get(handles.defespaco_exelement2_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_exelement2_elemento,'Value',1)
            set(handles.defespaco_exelement2_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_exelement2_elemento,'Value',1)
            end
        end
    end
end

```



```

        set(handles.defespaco_exelement2_elemento, 'String', '(Selecione)')
    else
        set(handles.defespaco_exelement2_elemento, 'Value', 1)
        set(handles.defespaco_exelement2_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.paredе.name})
    end
    case 3 %Pavimento

        if isempty(projecto(carproj).elementos.pavimento)==1
            set(handles.defespaco_exelement2_elemento, 'Value', 1)
            set(handles.defespaco_exelement2_elemento, 'String', '(Selecione)')
        else
            set(handles.defespaco_exelement2_elemento, 'Value', 1)
            set(handles.defespaco_exelement2_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.pavimento.name})
        end
    case 4 %Cobertura

        if isempty(projecto(carproj).elementos.cobertura)==1
            set(handles.defespaco_exelement2_elemento, 'Value', 1)
            set(handles.defespaco_exelement2_elemento, 'String', '(Selecione)')
        else
            set(handles.defespaco_exelement2_elemento, 'Value', 1)
            set(handles.defespaco_exelement2_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.cobertura.name})
        end
    case 5 %Teto

        if isempty(projecto(carproj).elementos.teto)==1
            set(handles.defespaco_exelement2_elemento, 'Value', 1)
            set(handles.defespaco_exelement2_elemento, 'String', '(Selecione)')
        else
            set(handles.defespaco_exelement2_elemento, 'Value', 1)
            set(handles.defespaco_exelement2_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.teto.name})
        end

    case 6 %Janela
        set(handles.defespaco_exelement2_elemento, 'Value', 1)
        set(handles.defespaco_exelement2_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).bdados.janela.name})
    case 7 %Porta
        set(handles.defespaco_exelement2_elemento, 'Value', 1)
        set(handles.defespaco_exelement2_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).bdados.porta.name})
    end
end

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement2_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement2_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

end

% --- Executes on selection change in defespaco_exelement2_elemento.
function defespaco_exelement2_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement2_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns ✓
defespaco_exelement2_elemento contents as cell array
%         contents{get(hObject,'Value')} returns selected item from ✓
defespaco_exelement2_elemento

% --- Executes during object creation, after setting all properties.
function defespaco_exelement2_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement2_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement2_quantidade_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement2_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement2_quantidade ✓
as text
%         str2double(get(hObject,'String')) returns contents of ✓
defespaco_exelement2_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement2_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement2_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement2_area_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement2_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of defespaco_exelement2_area as text
%         str2double(get(hObject,'String')) returns contents of
defespaco_exelement2_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement2_area_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement2_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_exelement4_tipo.
function defespaco_exelement4_tipo_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement4_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_exelement4_tipo
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
defespaco_exelement4_tipo
load projecto.mat
global carproj

if get(handles.defespaco_exelement4_check,'Value')==1

    switch get(handles.defespaco_exelement4_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_exelement4_elemento,'Value',1)
            set(handles.defespaco_exelement4_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_exelement4_elemento,'Value',1)
                set(handles.defespaco_exelement4_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement4_elemento,'Value',1)
                set(handles.defespaco_exelement4_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.parede.nome})
            end
        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_exelement4_elemento,'Value',1)
                set(handles.defespaco_exelement4_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement4_elemento,'Value',1)
                set(handles.defespaco_exelement4_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.pavimento.nome})
            end
    end
end

```

```

case 4 %Cobertura

    if isempty(projecto(carproj).elementos.cobertura)==1
        set(handles.defespaco_exelement4_elemento, 'Value', 1)
        set(handles.defespaco_exelement4_elemento, 'String', '(Selecione)')
    else
        set(handles.defespaco_exelement4_elemento, 'Value', 1)
        set(handles.defespaco_exelement4_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.cobertura.nome})
    end

case 5 %Teto

    if isempty(projecto(carproj).elementos.teto)==1
        set(handles.defespaco_exelement4_elemento, 'Value', 1)
        set(handles.defespaco_exelement4_elemento, 'String', '(Selecione)')
    else
        set(handles.defespaco_exelement4_elemento, 'Value', 1)
        set(handles.defespaco_exelement4_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.teto.nome})
    end

case 6 %Janela
    set(handles.defespaco_exelement4_elemento, 'Value', 1)
    set(handles.defespaco_exelement4_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).bdados.janela.nome})
case 7 %Porta
    set(handles.defespaco_exelement4_elemento, 'Value', 1)
    set(handles.defespaco_exelement4_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).bdados.porta.nome})
end
end

% --- Executes during object creation, after setting all properties.
function defespaco_exelement4_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement4_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on selection change in defespaco_exelement4_elemento.
function defespaco_exelement4_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement4_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String')) returns ✓
defespaco_exelement4_elemento contents as cell array
%         contents{get(hObject, 'Value')} returns selected item from ✓
defespaco_exelement4_elemento

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement4_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement4_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement4_quantidade_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement4_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement4_quantidade
as text
%         str2double(get(hObject,'String')) returns contents of
defespaco_exelement4_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement4_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement4_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement4_area_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement4_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement4_area as text
%         str2double(get(hObject,'String')) returns contents of
defespaco_exelement4_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement4_area_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement4_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_exelement5_tipo.
function defespaco_exelement5_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement5_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_exelement5_tipo
contents as cell array
%           contents{get(hObject,'Value')} returns selected item from
defespaco_exelement5_tipo
load projecto.mat
global carproj

if get(handles.defespaco_exelement5_check,'Value')==1

    switch get(handles.defespaco_exelement5_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_exelement5_elemento,'Value',1)
            set(handles.defespaco_exelement5_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_exelement5_elemento,'Value',1)
                set(handles.defespaco_exelement5_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement5_elemento,'Value',1)
                set(handles.defespaco_exelement5_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.parede.nome})
            end

        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_exelement5_elemento,'Value',1)
                set(handles.defespaco_exelement5_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement5_elemento,'Value',1)
                set(handles.defespaco_exelement5_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.pavimento.nome})
            end

        case 4 %Cobertura

            if isempty(projecto(carproj).elementos.cobertura)==1
                set(handles.defespaco_exelement5_elemento,'Value',1)
                set(handles.defespaco_exelement5_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement5_elemento,'Value',1)
                set(handles.defespaco_exelement5_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.cobertura.nome})
            end

        case 5 %Teto

```

```

        if isempty(projecto(carproj).elementos.teto)==1
            set(handles.defespaco_exelement5_elemento,'Value',1)
            set(handles.defespaco_exelement5_elemento,'String','(Selecione)')
        else
            set(handles.defespaco_exelement5_elemento,'Value',1)
            set(handles.defespaco_exelement5_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.teto.nome})
        end
        case 6 %Janela
            set(handles.defespaco_exelement5_elemento,'Value',1)
            set(handles.defespaco_exelement5_elemento,'String',{'(Selecione)',↵
projecto(carproj).bdados.janela.nome})
        case 7 %Porta
            set(handles.defespaco_exelement5_elemento,'Value',1)
            set(handles.defespaco_exelement5_elemento,'String',{'(Selecione)',↵
projecto(carproj).bdados.porta.nome})
        end
    end
end

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement5_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement5_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get↵
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in defespaco_exelement5_elemento.
function defespaco_exelement5_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement5_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject,'String')) returns↵
defespaco_exelement5_elemento contents as cell array
%       contents{get(hObject,'Value')} returns selected item from↵
defespaco_exelement5_elemento

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement5_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement5_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get↵
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function defespaco_exelement5_quantidade_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement5_quantidade (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement5_quantidade
as text
%           str2double(get(hObject,'String')) returns contents of
defespaco_exelement5_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement5_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement5_quantidade (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement5_area_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement5_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement5_area as text
%           str2double(get(hObject,'String')) returns contents of
defespaco_exelement5_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement5_area_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement5_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_exelement6_tipo.
function defespaco_exelement6_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement6_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_exelement6_tipo
contents as cell array

```



```

%         contents{get(hObject,'Value')} returns selected item from✓
defespaco_exelement6_tipo
load projecto.mat
global carproj

if get(handles.defespaco_exelement6_check,'Value')==1

    switch get(handles.defespaco_exelement6_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_exelement6_elemento,'Value',1)
            set(handles.defespaco_exelement6_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_exelement6_elemento,'Value',1)
                set(handles.defespaco_exelement6_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement6_elemento,'Value',1)
                set(handles.defespaco_exelement6_elemento,'String',{'(Selezione)',✓
projecto(carproj).elementos.parede.nome})
            end

        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_exelement6_elemento,'Value',1)
                set(handles.defespaco_exelement6_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement6_elemento,'Value',1)
                set(handles.defespaco_exelement6_elemento,'String',{'(Selezione)',✓
projecto(carproj).elementos.pavimento.nome})
            end

        case 4 %Cobertura

            if isempty(projecto(carproj).elementos.cobertura)==1
                set(handles.defespaco_exelement6_elemento,'Value',1)
                set(handles.defespaco_exelement6_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement6_elemento,'Value',1)
                set(handles.defespaco_exelement6_elemento,'String',{'(Selezione)',✓
projecto(carproj).elementos.cobertura.nome})
            end

        case 5 %Teto

            if isempty(projecto(carproj).elementos.teto)==1
                set(handles.defespaco_exelement6_elemento,'Value',1)
                set(handles.defespaco_exelement6_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement6_elemento,'Value',1)
                set(handles.defespaco_exelement6_elemento,'String',{'(Selezione)',✓
projecto(carproj).elementos.teto.nome})
            end

        case 6 %Janela
            set(handles.defespaco_exelement6_elemento,'Value',1)
            set(handles.defespaco_exelement6_elemento,'String',{'(Selezione)',✓
projecto(carproj).bdados.janela.nome})

```

```

        case 7 %Porta
            set(handles.defespaco_exelement6_elemento, 'Value', 1)
            set(handles.defespaco_exelement6_elemento, 'String', {'(Selecione)',
projecto(carproj).bdados.porta.nome})
        end
    end

% --- Executes during object creation, after setting all properties.
function defespaco_exelement6_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement6_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on selection change in defespaco_exelement6_elemento.
function defespaco_exelement6_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement6_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String')) returns
defespaco_exelement6_elemento contents as cell array
%         contents{get(hObject, 'Value')} returns selected item from
defespaco_exelement6_elemento

% --- Executes during object creation, after setting all properties.
function defespaco_exelement6_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement6_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function defespaco_exelement6_quantidade_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement6_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of defespaco_exelement6_quantidade
as text
%         str2double(get(hObject, 'String')) returns contents of
defespaco_exelement6_quantidade as a double

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement6_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement6_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement6_area_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement6_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement6_area as text
%         str2double(get(hObject,'String')) returns contents of
defespaco_exelement6_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement6_area_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement6_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_exelement7_tipo.
function defespaco_exelement7_tipo_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement7_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_exelement7_tipo
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
defespaco_exelement7_tipo
load projecto.mat
global carproj

if get(handles.defespaco_exelement7_check,'Value')==1

    switch get(handles.defespaco_exelement7_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_exelement7_elemento,'Value',1)
            set(handles.defespaco_exelement7_elemento,'String','(Selezione)')
        case 2 %Parede

```

```

        if isempty(projecto(carproj).elementos.paredes)==1
            set(handles.defespaco_exelement7_elemento,'Value',1)
            set(handles.defespaco_exelement7_elemento,'String','(Selecione)')
        else
            set(handles.defespaco_exelement7_elemento,'Value',1)
            set(handles.defespaco_exelement7_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.paredes.nome})
        end
        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_exelement7_elemento,'Value',1)
                set(handles.defespaco_exelement7_elemento,'String','(Selecione)')
            else
                set(handles.defespaco_exelement7_elemento,'Value',1)
                set(handles.defespaco_exelement7_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.pavimento.nome})
            end
            case 4 %Cobertura

                if isempty(projecto(carproj).elementos.cobertura)==1
                    set(handles.defespaco_exelement7_elemento,'Value',1)
                    set(handles.defespaco_exelement7_elemento,'String','(Selecione)')
                else
                    set(handles.defespaco_exelement7_elemento,'Value',1)
                    set(handles.defespaco_exelement7_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.cobertura.nome})
                end
                case 5 %Teto

                    if isempty(projecto(carproj).elementos.teto)==1
                        set(handles.defespaco_exelement7_elemento,'Value',1)
                        set(handles.defespaco_exelement7_elemento,'String','(Selecione)')
                    else
                        set(handles.defespaco_exelement7_elemento,'Value',1)
                        set(handles.defespaco_exelement7_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.teto.nome})
                    end

                    case 6 %Janela
                        set(handles.defespaco_exelement7_elemento,'Value',1)
                        set(handles.defespaco_exelement7_elemento,'String',{'(Selecione)',↵
projecto(carproj).bdados.janela.nome})
                    case 7 %Porta
                        set(handles.defespaco_exelement7_elemento,'Value',1)
                        set(handles.defespaco_exelement7_elemento,'String',{'(Selecione)',↵
projecto(carproj).bdados.porta.nome})
                    end
                end
            end
        end
    end
end

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement7_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement7_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_exelement7_elemento.
function defespaco_exelement7_elemento_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement7_elemento (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
defespaco_exelement7_elemento contents as cell array
%           contents{get(hObject,'Value')} returns selected item from
defespaco_exelement7_elemento

% --- Executes during object creation, after setting all properties.
function defespaco_exelement7_elemento_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement7_elemento (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement7_quantidade_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement7_quantidade (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement7_quantidade
as text
%           str2double(get(hObject,'String')) returns contents of
defespaco_exelement7_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement7_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement7_quantidade (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement7_area_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to defespaco_exelement7_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement7_area as text
%          str2double(get(hObject,'String')) returns contents of
defespaco_exelement7_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement7_area_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement7_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_exelement8_tipo.
function defespaco_exelement8_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement8_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_exelement8_tipo
contents as cell array
%          contents{get(hObject,'Value')} returns selected item from
defespaco_exelement8_tipo
load projecto.mat
global carproj

if get(handles.defespaco_exelement8_check,'Value')==1

    switch get(handles.defespaco_exelement8_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_exelement8_elemento,'Value',1)
            set(handles.defespaco_exelement8_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_exelement8_elemento,'Value',1)
                set(handles.defespaco_exelement8_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement8_elemento,'Value',1)
                set(handles.defespaco_exelement8_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.parede.nome})
            end
        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_exelement8_elemento,'Value',1)
                set(handles.defespaco_exelement8_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement8_elemento,'Value',1)

```

```

        set(handles.defespaco_exelement8_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.pavimento.nome})
    end
    case 4 %Cobertura

        if isempty(projecto(carproj).elementos.cobertura)==1
            set(handles.defespaco_exelement8_elemento, 'Value', 1)
            set(handles.defespaco_exelement8_elemento, 'String', '(Selecione)')
        else
            set(handles.defespaco_exelement8_elemento, 'Value', 1)
            set(handles.defespaco_exelement8_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.cobertura.nome})
        end
    case 5 %Teto

        if isempty(projecto(carproj).elementos.teto)==1
            set(handles.defespaco_exelement8_elemento, 'Value', 1)
            set(handles.defespaco_exelement8_elemento, 'String', '(Selecione)')
        else
            set(handles.defespaco_exelement8_elemento, 'Value', 1)
            set(handles.defespaco_exelement8_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.teto.nome})
        end
    case 6 %Janela
        set(handles.defespaco_exelement8_elemento, 'Value', 1)
        set(handles.defespaco_exelement8_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).bdados.janela.nome})
    case 7 %Porta
        set(handles.defespaco_exelement8_elemento, 'Value', 1)
        set(handles.defespaco_exelement8_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).bdados.porta.nome})
    end
end

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement8_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement8_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

% --- Executes on selection change in defespaco_exelement8_elemento.
function defespaco_exelement8_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement8_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject, 'String')) returns ✓
defespaco_exelement8_elemento contents as cell array
% contents{get(hObject, 'Value')} returns selected item from ✓
defespaco_exelement8_elemento

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement8_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement8_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement8_quantidade_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement8_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement8_quantidade
as text
%         str2double(get(hObject,'String')) returns contents of
defespaco_exelement8_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement8_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement8_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement8_area_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement8_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement8_area as text
%         str2double(get(hObject,'String')) returns contents of
defespaco_exelement8_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement8_area_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement8_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

```



```

% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_exelement3_tipo.
function defespaco_exelement3_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement3_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defespaco_exelement3_tipo
contents as cell array
%      contents{get(hObject,'Value')} returns selected item from
defespaco_exelement3_tipo
load projecto.mat
global carproj

if get(handles.defespaco_exelement3_check,'Value')==1

    switch get(handles.defespaco_exelement3_tipo,'Value')

        case 1 %Selezione

            set(handles.defespaco_exelement3_elemento,'Value',1)
            set(handles.defespaco_exelement3_elemento,'String','(Selezione)')

        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_exelement3_elemento,'Value',1)
                set(handles.defespaco_exelement3_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement3_elemento,'Value',1)
                set(handles.defespaco_exelement3_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.parede.nome})
            end

        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_exelement3_elemento,'Value',1)
                set(handles.defespaco_exelement3_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement3_elemento,'Value',1)
                set(handles.defespaco_exelement3_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.pavimento.nome})
            end

        case 4 %Cobertura

            if isempty(projecto(carproj).elementos.cobertura)==1
                set(handles.defespaco_exelement3_elemento,'Value',1)
                set(handles.defespaco_exelement3_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_exelement3_elemento,'Value',1)
                set(handles.defespaco_exelement3_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.cobertura.nome})
            end

        case 5 %Teto

```

```

        if isempty(projecto(carproj).elementos.teto)==1
            set(handles.defespaco_exelement3_elemento,'Value',1)
            set(handles.defespaco_exelement3_elemento,'String','(Selecione)')
        else
            set(handles.defespaco_exelement3_elemento,'Value',1)
            set(handles.defespaco_exelement3_elemento,'String',{'(Selecione)',✓
projecto(carproj).elementos.teto.nome})
        end
        case 6 %Janela
            set(handles.defespaco_exelement3_elemento,'Value',1)
            set(handles.defespaco_exelement3_elemento,'String',{'(Selecione)',✓
projecto(carproj).bdados.janela.nome})
        case 7 %Porta
            set(handles.defespaco_exelement3_elemento,'Value',1)
            set(handles.defespaco_exelement3_elemento,'String',{'(Selecione)',✓
projecto(carproj).bdados.porta.nome})
        end
    end
end

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement3_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement3_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in defespaco_exelement3_elemento.
function defespaco_exelement3_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement3_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns✓
defespaco_exelement3_elemento contents as cell array
%         contents{get(hObject,'Value')} returns selected item from✓
defespaco_exelement3_elemento

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_exelement3_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_exelement3_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function defespaco_exelement3_quantidade_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement3_quantidade (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement3_quantidade
as text
%           str2double(get(hObject,'String')) returns contents of
defespaco_exelement3_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement3_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement3_quantidade (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_exelement3_area_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement3_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_exelement3_area as text
%           str2double(get(hObject,'String')) returns contents of
defespaco_exelement3_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_exelement3_area_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_exelement3_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_intelement2_tipo.
function defespaco_intelement2_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement2_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
defespaco_intelement2_tipo contents as cell array
%           contents{get(hObject,'Value')} returns selected item from

```

```

defespaco_intelement2_tipo
load projecto.mat
global carproj

if get(handles.defespaco_intelement2_check, 'Value')==1

    switch get(handles.defespaco_intelement2_tipo, 'Value')

        case 1 %Selezione
            set(handles.defespaco_intelement2_elemento, 'Value', 1)
            set(handles.defespaco_intelement2_elemento, 'String', '(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_intelement2_elemento, 'Value', 1)
                set(handles.defespaco_intelement2_elemento, 'String', '(Selezione)')
            else
                set(handles.defespaco_intelement2_elemento, 'Value', 1)
                set(handles.defespaco_intelement2_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).elementos.parede.nome})
            end

        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_intelement2_elemento, 'Value', 1)
                set(handles.defespaco_intelement2_elemento, 'String', '(Selezione)')
            else
                set(handles.defespaco_intelement2_elemento, 'Value', 1)
                set(handles.defespaco_intelement2_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).elementos.pavimento.nome})
            end

        case 4 %Cobertura

            if isempty(projecto(carproj).elementos.cobertura)==1
                set(handles.defespaco_intelement2_elemento, 'Value', 1)
                set(handles.defespaco_intelement2_elemento, 'String', '(Selezione)')
            else
                set(handles.defespaco_intelement2_elemento, 'Value', 1)
                set(handles.defespaco_intelement2_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).elementos.cobertura.nome})
            end

        case 5 %Teto

            if isempty(projecto(carproj).elementos.teto)==1
                set(handles.defespaco_intelement2_elemento, 'Value', 1)
                set(handles.defespaco_intelement2_elemento, 'String', '(Selezione)')
            else
                set(handles.defespaco_intelement2_elemento, 'Value', 1)
                set(handles.defespaco_intelement2_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).elementos.teto.nome})
            end

        case 6 %Janela
            set(handles.defespaco_intelement2_elemento, 'Value', 1)
            set(handles.defespaco_intelement2_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).bdados.janela.nome})
        case 7 %Porta

```

```

        set(handles.defespaco_intelement2_elemento, 'Value', 1)
        set(handles.defespaco_intelement2_elemento, 'String', {'(Selecione)',
projecto(carproj).bdados.porta.nome})
    end
end

% --- Executes during object creation, after setting all properties.
function defespaco_intelement2_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement2_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on selection change in defespaco_intelement2_elemento.
function defespaco_intelement2_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement2_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String')) returns
defespaco_intelement2_elemento contents as cell array
%         contents{get(hObject, 'Value')} returns selected item from
defespaco_intelement2_elemento

% --- Executes during object creation, after setting all properties.
function defespaco_intelement2_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement2_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function defespaco_intelement2_quantidade_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement2_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of defespaco_intelement2_quantidade
as text
%         str2double(get(hObject, 'String')) returns contents of
defespaco_intelement2_quantidade as a double

% --- Executes during object creation, after setting all properties.

```

```

function defespaco_intelement2_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement2_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function defespaco_intelement2_area_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement2_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement2_area as
text
%         str2double(get(hObject,'String')) returns contents of
defespaco_intelement2_area as a double

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement2_area_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement2_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function defespaco_intelement2_temp_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement2_temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement2_temp as
text
%         str2double(get(hObject,'String')) returns contents of
defespaco_intelement2_temp as a double

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement2_temp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement2_temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(

```

```

(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_intelement3_tipo.
function defespaco_intelement3_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement3_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns ✓
defespaco_intelement3_tipo contents as cell array
%           contents{get(hObject,'Value')} returns selected item from ✓
defespaco_intelement3_tipo
load projecto.mat
global carproj

if get(handles.defespaco_intelement3_check,'Value')==1

    switch get(handles.defespaco_intelement3_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_intelement3_elemento,'Value',1)
            set(handles.defespaco_intelement3_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_intelement3_elemento,'Value',1)
                set(handles.defespaco_intelement3_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement3_elemento,'Value',1)
                set(handles.defespaco_intelement3_elemento,'String',{'(Selezione)',✓
projecto(carproj).elementos.parede.nome})
            end

        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_intelement3_elemento,'Value',1)
                set(handles.defespaco_intelement3_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement3_elemento,'Value',1)
                set(handles.defespaco_intelement3_elemento,'String',{'(Selezione)',✓
projecto(carproj).elementos.pavimento.nome})
            end

        case 4 %Cobertura

            if isempty(projecto(carproj).elementos.cobertura)==1
                set(handles.defespaco_intelement3_elemento,'Value',1)
                set(handles.defespaco_intelement3_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement3_elemento,'Value',1)
                set(handles.defespaco_intelement3_elemento,'String',{'(Selezione)',✓
projecto(carproj).elementos.cobertura.nome})
            end

        case 5 %Teto

```

```

        if isempty(projecto(carproj).elementos.teto)==1
            set(handles.defespaco_intelement3_elemento,'Value',1)
            set(handles.defespaco_intelement3_elemento,'String','(Selecione)')
        else
            set(handles.defespaco_intelement3_elemento,'Value',1)
            set(handles.defespaco_intelement3_elemento,'String',{'(Selecione)'},↵
projecto(carproj).elementos.teto.nome})
        end

        case 6 %Janela
            set(handles.defespaco_intelement3_elemento,'Value',1)
            set(handles.defespaco_intelement3_elemento,'String',{'(Selecione)'},↵
projecto(carproj).bdados.janela.nome})
        case 7 %Porta
            set(handles.defespaco_intelement3_elemento,'Value',1)
            set(handles.defespaco_intelement3_elemento,'String',{'(Selecione)'},↵
projecto(carproj).bdados.porta.nome})
        end
    end
end

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement3_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement3_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get↵
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in defespaco_intelement3_elemento.
function defespaco_intelement3_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement3_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns↵
defespaco_intelement3_elemento contents as cell array
%         contents{get(hObject,'Value')} returns selected item from↵
defespaco_intelement3_elemento

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement3_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement3_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get↵
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



```

function defespaco_intelement3_quantidade_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement3_quantidade (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement3_quantidade
as text
%           str2double(get(hObject,'String')) returns contents of
defespaco_intelement3_quantidade as a double

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement3_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement3_quantidade (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function defespaco_intelement3_area_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement3_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement3_area as
text
%           str2double(get(hObject,'String')) returns contents of
defespaco_intelement3_area as a double

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement3_area_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement3_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function defespaco_intelement3_temp_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement3_temp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement3_temp as

```

```

text
%         str2double(get(hObject,'String')) returns contents of
defespaco_intelement3_temp as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement3_temp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement3_temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_intelement4_tipo.
function defespaco_intelement4_tipo_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement4_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
defespaco_intelement4_tipo contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
defespaco_intelement4_tipo
load projecto.mat
global carproj

if get(handles.defespaco_intelement4_check,'Value')==1

    switch get(handles.defespaco_intelement4_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_intelement4_elemento,'Value',1)
            set(handles.defespaco_intelement4_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_intelement4_elemento,'Value',1)
                set(handles.defespaco_intelement4_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement4_elemento,'Value',1)
                set(handles.defespaco_intelement4_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.parede.nome})
            end

        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_intelement4_elemento,'Value',1)
                set(handles.defespaco_intelement4_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement4_elemento,'Value',1)
                set(handles.defespaco_intelement4_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.pavimento.nome})
            end
    end
end

```

```

case 4 %Cobertura

    if isempty(projecto(carproj).elementos.cobertura)==1
        set(handles.defespaco_intelement4_elemento,'Value',1)
        set(handles.defespaco_intelement4_elemento,'String','(Selecione)')
    else
        set(handles.defespaco_intelement4_elemento,'Value',1)
        set(handles.defespaco_intelement4_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.cobertura.nome})
    end

case 5 %Teto

    if isempty(projecto(carproj).elementos.teto)==1
        set(handles.defespaco_intelement4_elemento,'Value',1)
        set(handles.defespaco_intelement4_elemento,'String','(Selecione)')
    else
        set(handles.defespaco_intelement4_elemento,'Value',1)
        set(handles.defespaco_intelement4_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.teto.nome})
    end

case 6 %Janela
    set(handles.defespaco_intelement4_elemento,'Value',1)
    set(handles.defespaco_intelement4_elemento,'String',{'(Selecione)',↵
projecto(carproj).bdados.janela.nome})

case 7 %Porta
    set(handles.defespaco_intelement4_elemento,'Value',1)
    set(handles.defespaco_intelement4_elemento,'String',{'(Selecione)',↵
projecto(carproj).bdados.porta.nome})
end
end

% --- Executes during object creation, after setting all properties.
function defespaco_intelement4_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement4_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get↵
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_intelement4_elemento.
function defespaco_intelement4_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement4_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns↵
defespaco_intelement4_elemento contents as cell array
%         contents{get(hObject,'Value')} returns selected item from↵
defespaco_intelement4_elemento

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement4_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement4_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement4_quantidade_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement4_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement4_quantidade
as text
%       str2double(get(hObject,'String')) returns contents of
defespaco_intelement4_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement4_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement4_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement4_area_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement4_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement4_area as
text
%       str2double(get(hObject,'String')) returns contents of
defespaco_intelement4_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement4_area_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement4_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement4_temp_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement4_temp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement4_temp as
text
%           str2double(get(hObject,'String')) returns contents of
defespaco_intelement4_temp as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement4_temp_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement4_temp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_intelement5_tipo.
function defespaco_intelement5_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement5_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
defespaco_intelement5_tipo contents as cell array
%           contents{get(hObject,'Value')} returns selected item from
defespaco_intelement5_tipo
load projecto.mat
global carproj

if get(handles.defespaco_intelement5_check,'Value')==1

    switch get(handles.defespaco_intelement5_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_intelement5_elemento,'Value',1)
            set(handles.defespaco_intelement5_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_intelement5_elemento,'Value',1)
                set(handles.defespaco_intelement5_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement5_elemento,'Value',1)
                set(handles.defespaco_intelement5_elemento,'String',{'(Selezione)',

```

```

projecto(carproj).elementos.parede.nome})
    end

    case 3 %Pavimento

        if isempty(projecto(carproj).elementos.pavimento)==1
            set(handles.defespaco_intelement5_elemento,'Value',1)
            set(handles.defespaco_intelement5_elemento,'String','(Selecione)')
        else
            set(handles.defespaco_intelement5_elemento,'Value',1)
            set(handles.defespaco_intelement5_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.pavimento.nome})
        end

    case 4 %Cobertura

        if isempty(projecto(carproj).elementos.cobertura)==1
            set(handles.defespaco_intelement5_elemento,'Value',1)
            set(handles.defespaco_intelement5_elemento,'String','(Selecione)')
        else
            set(handles.defespaco_intelement5_elemento,'Value',1)
            set(handles.defespaco_intelement5_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.cobertura.nome})
        end

    case 5 %Teto

        if isempty(projecto(carproj).elementos.teto)==1
            set(handles.defespaco_intelement5_elemento,'Value',1)
            set(handles.defespaco_intelement5_elemento,'String','(Selecione)')
        else
            set(handles.defespaco_intelement5_elemento,'Value',1)
            set(handles.defespaco_intelement5_elemento,'String',{'(Selecione)',↵
projecto(carproj).elementos.teto.nome})
        end

    case 6 %Janela
        set(handles.defespaco_intelement5_elemento,'Value',1)
        set(handles.defespaco_intelement5_elemento,'String',{'(Selecione)',↵
projecto(carproj).bdados.janela.nome})
    case 7 %Porta
        set(handles.defespaco_intelement5_elemento,'Value',1)
        set(handles.defespaco_intelement5_elemento,'String',{'(Selecione)',↵
projecto(carproj).bdados.porta.nome})
    end
end

% --- Executes during object creation, after setting all properties.
function defespaco_intelement5_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement5_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get↵
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in defespaco_intelement5_elemento.
function defespaco_intelement5_elemento_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement5_elemento (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns ✓
defespaco_intelement5_elemento contents as cell array
%           contents{get(hObject,'Value')} returns selected item from ✓
defespaco_intelement5_elemento

% --- Executes during object creation, after setting all properties.
function defespaco_intelement5_elemento_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement5_elemento (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement5_quantidade_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement5_quantidade (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement5_quantidade ✓
as text
%           str2double(get(hObject,'String')) returns contents of ✓
defespaco_intelement5_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement5_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement5_quantidade (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement5_area_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement5_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of defespaco_intelement5_area as✓
text
%         str2double(get(hObject,'String')) returns contents of✓
defespaco_intelement5_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement5_area_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement5_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement5_temp_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement5_temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement5_temp as✓
text
%         str2double(get(hObject,'String')) returns contents of✓
defespaco_intelement5_temp as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement5_temp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement5_temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_intelement6_tipo.
function defespaco_intelement6_tipo_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement6_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns✓
defespaco_intelement6_tipo contents as cell array
%         contents{get(hObject,'Value')} returns selected item from✓
defespaco_intelement6_tipo

load projecto.mat
global carproj

```



```

if get(handles.defespaco_intelement6_check, 'Value')==1

    switch get(handles.defespaco_intelement6_tipo, 'Value')

        case 1 %Selezione
            set(handles.defespaco_intelement6_elemento, 'Value', 1)
            set(handles.defespaco_intelement6_elemento, 'String', '(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_intelement6_elemento, 'Value', 1)
                set(handles.defespaco_intelement6_elemento, 'String', '(Selezione)')
            else
                set(handles.defespaco_intelement6_elemento, 'Value', 1)
                set(handles.defespaco_intelement6_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).elementos.parede.nome})
            end

        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_intelement6_elemento, 'Value', 1)
                set(handles.defespaco_intelement6_elemento, 'String', '(Selezione)')
            else
                set(handles.defespaco_intelement6_elemento, 'Value', 1)
                set(handles.defespaco_intelement6_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).elementos.pavimento.nome})
            end

        case 4 %Cobertura

            if isempty(projecto(carproj).elementos.cobertura)==1
                set(handles.defespaco_intelement6_elemento, 'Value', 1)
                set(handles.defespaco_intelement6_elemento, 'String', '(Selezione)')
            else
                set(handles.defespaco_intelement6_elemento, 'Value', 1)
                set(handles.defespaco_intelement6_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).elementos.cobertura.nome})
            end

        case 5 %Teto

            if isempty(projecto(carproj).elementos.teto)==1
                set(handles.defespaco_intelement6_elemento, 'Value', 1)
                set(handles.defespaco_intelement6_elemento, 'String', '(Selezione)')
            else
                set(handles.defespaco_intelement6_elemento, 'Value', 1)
                set(handles.defespaco_intelement6_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).elementos.teto.nome})
            end

        case 6 %Janela
            set(handles.defespaco_intelement6_elemento, 'Value', 1)
            set(handles.defespaco_intelement6_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).bdados.janela.nome})
        case 7 %Porta
            set(handles.defespaco_intelement6_elemento, 'Value', 1)
            set(handles.defespaco_intelement6_elemento, 'String', {'(Selezione)', ✓
projecto(carproj).bdados.porta.nome})
        end
    end
end

```

end

```
% --- Executes during object creation, after setting all properties.
function defespaco_intelement6_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement6_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on selection change in defespaco_intelement6_elemento.
function defespaco_intelement6_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement6_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: contents = cellstr(get(hObject,'String')) returns
defespaco_intelement6_elemento contents as cell array
%       contents{get(hObject,'Value')} returns selected item from
defespaco_intelement6_elemento
```

```
% --- Executes during object creation, after setting all properties.
function defespaco_intelement6_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement6_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function defespaco_intelement6_quantidade_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement6_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of defespaco_intelement6_quantidade
as text
%       str2double(get(hObject,'String')) returns contents of
defespaco_intelement6_quantidade as a double
```

```
% --- Executes during object creation, after setting all properties.
function defespaco_intelement6_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement6_quantidade (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement6_area_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement6_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement6_area as
text
%      str2double(get(hObject,'String')) returns contents of
defespaco_intelement6_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement6_area_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement6_area (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement6_temp_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement6_temp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement6_temp as
text
%      str2double(get(hObject,'String')) returns contents of
defespaco_intelement6_temp as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement6_temp_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement6_temp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in defespaco_intelement7_tipo.
function defespaco_intelement7_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement7_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns ✓
defespaco_intelement7_tipo contents as cell array
%           contents{get(hObject,'Value')} returns selected item from ✓
defespaco_intelement7_tipo
load projecto.mat
global carproj

if get(handles.defespaco_intelement7_check,'Value')==1

    switch get(handles.defespaco_intelement7_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_intelement7_elemento,'Value',1)
            set(handles.defespaco_intelement7_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_intelement7_elemento,'Value',1)
                set(handles.defespaco_intelement7_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement7_elemento,'Value',1)
                set(handles.defespaco_intelement7_elemento,'String',{'(Selezione)', ✓
projecto(carproj).elementos.parede.nome})
            end

        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_intelement7_elemento,'Value',1)
                set(handles.defespaco_intelement7_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement7_elemento,'Value',1)
                set(handles.defespaco_intelement7_elemento,'String',{'(Selezione)', ✓
projecto(carproj).elementos.pavimento.nome})
            end

        case 4 %Cobertura

            if isempty(projecto(carproj).elementos.cobertura)==1
                set(handles.defespaco_intelement7_elemento,'Value',1)
                set(handles.defespaco_intelement7_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement7_elemento,'Value',1)
                set(handles.defespaco_intelement7_elemento,'String',{'(Selezione)', ✓
projecto(carproj).elementos.cobertura.nome})
            end

        case 5 %Teto

            if isempty(projecto(carproj).elementos.teto)==1
                set(handles.defespaco_intelement7_elemento,'Value',1)
                set(handles.defespaco_intelement7_elemento,'String','(Selezione)')
            else

```

```

        set(handles.defespaco_intelement7_elemento, 'Value', 1)
        set(handles.defespaco_intelement7_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.teto.nome})
    end

    case 6 %Janela
        set(handles.defespaco_intelement7_elemento, 'Value', 1)
        set(handles.defespaco_intelement7_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).bdados.janela.nome})
    case 7 %Porta
        set(handles.defespaco_intelement7_elemento, 'Value', 1)
        set(handles.defespaco_intelement7_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).bdados.porta.nome})
    end
end
end

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement7_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement7_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

% --- Executes on selection change in defespaco_intelement7_elemento.
function defespaco_intelement7_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement7_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject, 'String')) returns ✓
defespaco_intelement7_elemento contents as cell array
%       contents{get(hObject, 'Value')} returns selected item from ✓
defespaco_intelement7_elemento

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement7_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement7_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

function defespaco_intelement7_quantidade_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement7_quantidade (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement7_quantidade as text
% str2double(get(hObject,'String')) returns contents of defespaco_intelement7_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement7_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject handle to defespaco_intelement7_quantidade (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement7_area_Callback(hObject, eventdata, handles)
% hObject handle to defespaco_intelement7_area (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement7_area as text
% str2double(get(hObject,'String')) returns contents of defespaco_intelement7_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement7_area_CreateFcn(hObject, eventdata, handles)
% hObject handle to defespaco_intelement7_area (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement7_temp_Callback(hObject, eventdata, handles)
% hObject handle to defespaco_intelement7_temp (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement7_temp as text
% str2double(get(hObject,'String')) returns contents of defespaco_intelement7_temp as a double

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement7_temp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement7_temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in defespaco_intelement8_tipo.
function defespaco_intelement8_tipo_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement8_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
defespaco_intelement8_tipo contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
defespaco_intelement8_tipo
load projecto.mat
global carproj

if get(handles.defespaco_intelement8_check,'Value')==1

    switch get(handles.defespaco_intelement8_tipo,'Value')

        case 1 %Selezione
            set(handles.defespaco_intelement8_elemento,'Value',1)
            set(handles.defespaco_intelement8_elemento,'String','(Selezione)')
        case 2 %Parede

            if isempty(projecto(carproj).elementos.parede)==1
                set(handles.defespaco_intelement8_elemento,'Value',1)
                set(handles.defespaco_intelement8_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement8_elemento,'Value',1)
                set(handles.defespaco_intelement8_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.parede.nome})
            end

        case 3 %Pavimento

            if isempty(projecto(carproj).elementos.pavimento)==1
                set(handles.defespaco_intelement8_elemento,'Value',1)
                set(handles.defespaco_intelement8_elemento,'String','(Selezione)')
            else
                set(handles.defespaco_intelement8_elemento,'Value',1)
                set(handles.defespaco_intelement8_elemento,'String',{'(Selezione)',
projecto(carproj).elementos.pavimento.nome})
            end

        case 4 %Cobertura

            if isempty(projecto(carproj).elementos.cobertura)==1

```

```

        set(handles.defespaco_intelement8_elemento, 'Value', 1)
        set(handles.defespaco_intelement8_elemento, 'String', '(Selecione)')
    else
        set(handles.defespaco_intelement8_elemento, 'Value', 1)
        set(handles.defespaco_intelement8_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.cobertura.nome})
    end

    case 5 %Teto

        if isempty(projecto(carproj).elementos.teto)==1
            set(handles.defespaco_intelement8_elemento, 'Value', 1)
            set(handles.defespaco_intelement8_elemento, 'String', '(Selecione)')
        else
            set(handles.defespaco_intelement8_elemento, 'Value', 1)
            set(handles.defespaco_intelement8_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).elementos.teto.nome})
        end

    case 6 %Janela
        set(handles.defespaco_intelement8_elemento, 'Value', 1)
        set(handles.defespaco_intelement8_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).bdados.janela.nome})
    case 7 %Porta
        set(handles.defespaco_intelement8_elemento, 'Value', 1)
        set(handles.defespaco_intelement8_elemento, 'String', {'(Selecione)', ✓
projecto(carproj).bdados.porta.nome})
    end
end

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement8_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement8_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

% --- Executes on selection change in defespaco_intelement8_elemento.
function defespaco_intelement8_elemento_Callback(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement8_elemento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject, 'String')) returns ✓
defespaco_intelement8_elemento contents as cell array
%       contents{get(hObject, 'Value')} returns selected item from ✓
defespaco_intelement8_elemento

```

```

% --- Executes during object creation, after setting all properties.
function defespaco_intelement8_elemento_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defespaco_intelement8_elemento (see GCBO)

```



```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement8_quantidade_Callback(hObject, eventdata, handles)
% hObject handle to defespaco_intelement8_quantidade (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement8_quantidade
as text
% str2double(get(hObject,'String')) returns contents of
defespaco_intelement8_quantidade as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement8_quantidade_CreateFcn(hObject, eventdata, handles)
% hObject handle to defespaco_intelement8_quantidade (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_intelement8_area_Callback(hObject, eventdata, handles)
% hObject handle to defespaco_intelement8_area (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement8_area as
text
% str2double(get(hObject,'String')) returns contents of
defespaco_intelement8_area as a double

% --- Executes during object creation, after setting all properties.
function defespaco_intelement8_area_CreateFcn(hObject, eventdata, handles)
% hObject handle to defespaco_intelement8_area (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

end

```
function defespaco_intelement8_temp_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement8_temp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_intelement8_temp as
text
%           str2double(get(hObject,'String')) returns contents of
defespaco_intelement8_temp as a double
```

```
% --- Executes during object creation, after setting all properties.
function defespaco_intelement8_temp_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_intelement8_temp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in defespaco_guardar.
```

```
function defespaco_guardar_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_guardar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
choice = questdlg('Deseja adicionar o espaco?', 'Adicionar
Espaço', 'Sim', 'Não', 'Cancelar', 'Cancelar');
```

```
switch choice
```

```
    case 'Sim'
```

```
        load projecto.mat
        global carproj
```

```
        k = length(projecto(carproj).espaco)+1;
```

```
        projecto(carproj).espaco(k).nome = get(handles.defespaco_nome, 'String');
        projecto(carproj).espaco(k).temp_interior = get(handles.
```

```
defespaco_ti, 'String');
```

```
        projecto(carproj).espaco(k).hum_interior = get(handles.
```

```
defespaco_hi, 'String');
```

```
        projecto(carproj).espaco(k).area_espaco = get(handles.
```

```
defespaco_area, 'String');
```

```
        % Ocupação
```

```
        switch get(handles.defespaco_ocupunid, 'Value')
```

```
            case 1
```

```
            case 2 % Ocupação
```

```

        projecto(carproj).espaco(k).ocupacao.quant = get(handles.✓
defespaco_numocupacao, 'String');
        case 3 %Densidade de Ocupação
            projecto(carproj).espaco(k).ocupacao.quant = (str2double(get✓
(handles.defespaco_numocupacao, 'String'))^-1*str2double(get(handles.✓
defespaco_area, 'String'));
        end

        projecto(carproj).espaco(k).ocupacao.csensivel=get(handles.✓
defespaco_ocupsensivel, 'String');
        projecto(carproj).espaco(k).ocupacao.clatente=get(handles.✓
defespaco_ocuplatente, 'String');

        % Iluminação

        switch get(handles.defespaco_ilumunid, 'Value')
            case 1
            case 2
                projecto(carproj).espaco(k).ilumina = get(handles.✓
defespaco_ilupot, 'String');
            case 3 % Densidade de iluminação
                projecto(carproj).espaco(k).ilumina = str2double(get(handles.✓
defespaco_ilupot, 'String'))*str2double(get(handles.defespaco_area, 'String'));
            end

            % Equipamentos elétricos

            switch get(handles.defespaco_elepotunid, 'Value')
                case 1
                case 2
                    projecto(carproj).espaco(k).eletricos = get(handles.✓
defespaco_elepot, 'String');
                case 3 % Densidade de iluminação
                    projecto(carproj).espaco(k).eletricos= str2double(get(handles.✓
defespaco_elepot, 'String'))*str2double(get(handles.defespaco_area, 'String'));
                end

                %Cargas diversas

                projecto(carproj).espaco(k).diversas.sensivel=get(handles.✓
defespaco_disensivel, 'String');
                projecto(carproj).espaco(k).diversas.latente=get(handles.✓
defespaco_dilatente, 'String');

                % Infiltrações

                projecto(carproj).espaco(k).infiltracoes.caudal = get(handles.✓
defespaco_caudalar, 'String');
                projecto(carproj).espaco(k).infiltracoes.HR_extv = get(handles.✓
defespaco_hnext, 'String');
                projecto(carproj).espaco(k).infiltracoes.HR_exti = get(handles.✓
defespaco_hrint, 'String');

            save projecto.mat projecto

            % Ganhos de calor externos
            if get(handles.defespaco_exelement1_check, 'Value')==1

                x1=get(handles.defespaco_exelement1_tipo, 'Value');
                x2=get(handles.defespaco_exelement1_elemento, 'Value');

```

```

[y1,y2,y3]=elementos_espaco(x1,x2);

projecto(carproj).espaco(k).ganhos_ext(1).tipo = y1;
projecto(carproj).espaco(k).ganhos_ext(1).nome = y2;
projecto(carproj).espaco(k).ganhos_ext(1).u = y3;
projecto(carproj).espaco(k).ganhos_ext(1).quant = get(handles.✓
defespaco_exelement1_quantidade, 'String');
projecto(carproj).espaco(k).ganhos_ext(1).area = get(handles.✓
defespaco_exelement1_area, 'String');

clearvars x1 x2 y1 y2 y3

save projecto.mat projecto

end
if get(handles.defespaco_exelement2_check, 'Value')==1

x1=get(handles.defespaco_exelement2_tipo, 'Value');
x2=get(handles.defespaco_exelement2_elemento, 'Value');

[y1,y2,y3]=elementos_espaco(x1,x2);

projecto(carproj).espaco(k).ganhos_ext(2).tipo = y1;
projecto(carproj).espaco(k).ganhos_ext(2).nome = y2;
projecto(carproj).espaco(k).ganhos_ext(2).u = y3;
projecto(carproj).espaco(k).ganhos_ext(2).quant = get(handles.✓
defespaco_exelement2_quantidade, 'String');
projecto(carproj).espaco(k).ganhos_ext(2).area = get(handles.✓
defespaco_exelement2_area, 'String');

clearvars x1 x2 y1 y2 y3

save projecto.mat projecto

end
if get(handles.defespaco_exelement3_check, 'Value')==1

x1=get(handles.defespaco_exelement3_tipo, 'Value');
x2=get(handles.defespaco_exelement3_elemento, 'Value');

[y1,y2,y3]=elementos_espaco(x1,x2);

projecto(carproj).espaco(k).ganhos_ext(3).tipo = y1;
projecto(carproj).espaco(k).ganhos_ext(3).nome = y2;
projecto(carproj).espaco(k).ganhos_ext(3).u = y3;
projecto(carproj).espaco(k).ganhos_ext(3).quant = get(handles.✓
defespaco_exelement3_quantidade, 'String');
projecto(carproj).espaco(k).ganhos_ext(3).area = get(handles.✓
defespaco_exelement3_area, 'String');

clearvars x1 x2 y1 y2 y3

save projecto.mat projecto

end
if get(handles.defespaco_exelement4_check, 'Value')==1

x1=get(handles.defespaco_exelement4_tipo, 'Value');
x2=get(handles.defespaco_exelement4_elemento, 'Value');

[y1,y2,y3]=elementos_espaco(x1,x2);

```

```

        projecto(carproj).espaco(k).ganhos_ext(4).tipo = y1;
        projecto(carproj).espaco(k).ganhos_ext(4).nome = y2;
        projecto(carproj).espaco(k).ganhos_ext(4).u = y3;
        projecto(carproj).espaco(k).ganhos_ext(4).quant = get(handles.✓
defespaco_exelement4_quantidade, 'String');
        projecto(carproj).espaco(k).ganhos_ext(4).area = get(handles.✓
defespaco_exelement4_area, 'String');

        clearvars x1 x2 y1 y2 y3

        save projecto.mat projecto
end
if get(handles.defespaco_exelement5_check, 'Value')==1

        x1=get(handles.defespaco_exelement5_tipo, 'Value');
        x2=get(handles.defespaco_exelement5_elemento, 'Value');

        [y1,y2,y3]=elementos_espaco(x1,x2);

        projecto(carproj).espaco(k).ganhos_ext(5).tipo = y1;
        projecto(carproj).espaco(k).ganhos_ext(5).nome = y2;
        projecto(carproj).espaco(k).ganhos_ext(5).u = y3;
        projecto(carproj).espaco(k).ganhos_ext(5).quant = get(handles.✓
defespaco_exelement5_quantidade, 'String');
        projecto(carproj).espaco(k).ganhos_ext(5).area = get(handles.✓
defespaco_exelement5_area, 'String');

        clearvars x1 x2 y1 y2 y3

        save projecto.mat projecto
end
if get(handles.defespaco_exelement6_check, 'Value')==1
        x1=get(handles.defespaco_exelement6_tipo, 'Value');
        x2=get(handles.defespaco_exelement6_elemento, 'Value');

        [y1,y2,y3]=elementos_espaco(x1,x2);

        projecto(carproj).espaco(k).ganhos_ext(6).tipo = y1;
        projecto(carproj).espaco(k).ganhos_ext(6).nome = y2;
        projecto(carproj).espaco(k).ganhos_ext(6).u = y3;
        projecto(carproj).espaco(k).ganhos_ext(6).quant = get(handles.✓
defespaco_exelement6_quantidade, 'String');
        projecto(carproj).espaco(k).ganhos_ext(6).area = get(handles.✓
defespaco_exelement6_area, 'String');

        clearvars x1 x2 y1 y2 y3

        save projecto.mat projecto
end
if get(handles.defespaco_exelement7_check, 'Value')==1

        x1=get(handles.defespaco_exelement7_tipo, 'Value');
        x2=get(handles.defespaco_exelement7_elemento, 'Value');

        [y1,y2,y3]=elementos_espaco(x1,x2);

        projecto(carproj).espaco(k).ganhos_ext(7).tipo = y1;
        projecto(carproj).espaco(k).ganhos_ext(7).nome = y2;
        projecto(carproj).espaco(k).ganhos_ext(7).u = y3;
        projecto(carproj).espaco(k).ganhos_ext(7).quant = get(handles.✓

```

```

defespaco_exelement7_quantidade, 'String');
    projecto(carproj).espaco(k).ganhos_ext(7).area = get(handles.✓
defespaco_exelement7_area, 'String');

    clearvars x1 x2 y1 y2 y3

    save projecto.mat projecto

end
if get(handles.defespaco_exelement8_check, 'Value')==1

    x1=get(handles.defespaco_exelement8_tipo, 'Value');
    x2=get(handles.defespaco_exelement8_elemento, 'Value');

    [y1,y2,y3]=elementos_espaco(x1,x2);

    projecto(carproj).espaco(k).ganhos_ext(8).tipo = y1;
    projecto(carproj).espaco(k).ganhos_ext(8).nome = y2;
    projecto(carproj).espaco(k).ganhos_ext(8).u = y3;
    projecto(carproj).espaco(k).ganhos_ext(8).quant = get(handles.✓
defespaco_exelement8_quantidade, 'String');
    projecto(carproj).espaco(k).ganhos_ext(8).area = get(handles.✓
defespaco_exelement8_area, 'String');

    clearvars x1 x2 y1 y2 y3

    save projecto.mat projecto

end

save projecto.mat projecto
%%

% Ganhos de calor internos - Referente aos elementos

if get(handles.defespaco_intelement1_check, 'Value')==1

    x1=get(handles.defespaco_intelement1_tipo, 'Value');
    x2=get(handles.defespaco_intelement1_elemento, 'Value');

    [y1,y2,y3]=elementos_espaco(x1,x2);

    projecto(carproj).espaco(k).ganhos_int(1).tipo = y1;
    projecto(carproj).espaco(k).ganhos_int(1).nome = y2;
    projecto(carproj).espaco(k).ganhos_int(1).u = y3;
    projecto(carproj).espaco(k).ganhos_int(1).quant = get(handles.✓
defespaco_intelement1_quantidade, 'String');
    projecto(carproj).espaco(k).ganhos_int(1).area = get(handles.✓
defespaco_intelement1_area, 'String');
    projecto(carproj).espaco(k).ganhos_int(1).temp = get(handles.✓
defespaco_intelement1_temp, 'String');
    clearvars x1 x2 y1 y2 y3

    save projecto.mat projecto

end
if get(handles.defespaco_intelement2_check, 'Value')==1

    x1=get(handles.defespaco_intelement2_tipo, 'Value');
    x2=get(handles.defespaco_intelement2_elemento, 'Value');

    [y1,y2,y3]=elementos_espaco(x1,x2);

```

```

        projecto(carproj).espaco(k).ganhos_int(2).tipo = y1;
        projecto(carproj).espaco(k).ganhos_int(2).nome = y2;
        projecto(carproj).espaco(k).ganhos_int(2).u = y3;
        projecto(carproj).espaco(k).ganhos_int(2).quant = get(handles.✓
defespaco_intelement2_quantidade, 'String');
        projecto(carproj).espaco(k).ganhos_int(2).area = get(handles.✓
defespaco_intelement2_area, 'String');
        projecto(carproj).espaco(k).ganhos_int(2).temp = get(handles.✓
defespaco_intelement2_temp, 'String');

        clearvars x1 x2 y1 y2 y3

        save projecto.mat projecto

end
if get(handles.defespaco_intelement3_check, 'Value')==1

        x1=get(handles.defespaco_intelement3_tipo, 'Value');
        x2=get(handles.defespaco_intelement3_elemento, 'Value');

        [y1,y2,y3]=elementos_espaco(x1,x2);

        projecto(carproj).espaco(k).ganhos_int(3).tipo = y1;
        projecto(carproj).espaco(k).ganhos_int(3).nome = y2;
        projecto(carproj).espaco(k).ganhos_int(3).u = y3;
        projecto(carproj).espaco(k).ganhos_int(3).quant = get(handles.✓
defespaco_intelement3_quantidade, 'String');
        projecto(carproj).espaco(k).ganhos_int(3).area = get(handles.✓
defespaco_intelement3_area, 'String');
        projecto(carproj).espaco(k).ganhos_int(3).temp = get(handles.✓
defespaco_intelement3_temp, 'String');

        clearvars x1 x2 y1 y2 y3

        save projecto.mat projecto

end
if get(handles.defespaco_intelement4_check, 'Value')==1

        x1=get(handles.defespaco_intelement4_tipo, 'Value');
        x2=get(handles.defespaco_intelement4_elemento, 'Value');

        [y1,y2,y3]=elementos_espaco(x1,x2);

        projecto(carproj).espaco(k).ganhos_int(4).tipo = y1;
        projecto(carproj).espaco(k).ganhos_int(4).nome = y2;
        projecto(carproj).espaco(k).ganhos_int(4).u = y3;
        projecto(carproj).espaco(k).ganhos_int(4).quant = get(handles.✓
defespaco_intelement4_quantidade, 'String');
        projecto(carproj).espaco(k).ganhos_int(4).area = get(handles.✓
defespaco_intelement4_area, 'String');
        projecto(carproj).espaco(k).ganhos_int(4).temp = get(handles.✓
defespaco_intelement4_temp, 'String');

        clearvars x1 x2 y1 y2 y3

        save projecto.mat projecto

end
if get(handles.defespaco_intelement5_check, 'Value')==1

        x1=get(handles.defespaco_intelement5_tipo, 'Value');

```

```

x2=get(handles.defespaco_intelement5_elemento,'Value');

[y1,y2,y3]=elementos_espaco(x1,x2);

projecto(carproj).espaco(k).ganhos_int(5).tipo = y1;
projecto(carproj).espaco(k).ganhos_int(5).nome = y2;
projecto(carproj).espaco(k).ganhos_int(5).u = y3;
projecto(carproj).espaco(k).ganhos_int(5).quant = get(handles.✓
defespaco_intelement5_quantidade,'String');
projecto(carproj).espaco(k).ganhos_int(5).area = get(handles.✓
defespaco_intelement5_area,'String');
projecto(carproj).espaco(k).ganhos_int(5).temp = get(handles.✓
defespaco_intelement5_temp,'String');

clearvars x1 x2 y1 y2 y3

save projecto.mat projecto

end
if get(handles.defespaco_intelement6_check,'Value')==1
x1=get(handles.defespaco_intelement6_tipo,'Value');
x2=get(handles.defespaco_intelement6_elemento,'Value');

[y1,y2,y3]=elementos_espaco(x1,x2);

projecto(carproj).espaco(k).ganhos_int(6).tipo = y1;
projecto(carproj).espaco(k).ganhos_int(6).nome = y2;
projecto(carproj).espaco(k).ganhos_int(6).u = y3;
projecto(carproj).espaco(k).ganhos_int(6).quant = get(handles.✓
defespaco_intelement6_quantidade,'String');
projecto(carproj).espaco(k).ganhos_int(6).area = get(handles.✓
defespaco_intelement6_area,'String');
projecto(carproj).espaco(k).ganhos_int(6).temp = get(handles.✓
defespaco_intelement6_temp,'String');

clearvars x1 x2 y1 y2 y3

save projecto.mat projecto

end
if get(handles.defespaco_intelement7_check,'Value')==1

x1=get(handles.defespaco_intelement7_tipo,'Value');
x2=get(handles.defespaco_intelement7_elemento,'Value');

[y1,y2,y3]=elementos_espaco(x1,x2);

projecto(carproj).espaco(k).ganhos_int(7).tipo = y1;
projecto(carproj).espaco(k).ganhos_int(7).nome = y2;
projecto(carproj).espaco(k).ganhos_int(7).u = y3;
projecto(carproj).espaco(k).ganhos_int(7).quant = get(handles.✓
defespaco_intelement7_quantidade,'String');
projecto(carproj).espaco(k).ganhos_int(7).area = get(handles.✓
defespaco_intelement7_area,'String');
projecto(carproj).espaco(k).ganhos_int(7).temp = get(handles.✓
defespaco_intelement7_temp,'String');

clearvars x1 x2 y1 y2 y3

save projecto.mat projecto

end
if get(handles.defespaco_intelement8_check,'Value')==1

```



```

x1=get(handles.defespaco_intelement8_tipo,'Value');
x2=get(handles.defespaco_intelement8_elemento,'Value');

[y1,y2,y3]=elementos_espaco(x1,x2);

projecto(carproj).espaco(k).ganhos_int(8).tipo = y1;
projecto(carproj).espaco(k).ganhos_int(8).nome = y2;
projecto(carproj).espaco(k).ganhos_int(8).u = y3;
projecto(carproj).espaco(k).ganhos_int(8).quant = get(handles.✓
defespaco_intelement8_quantidade,'String');
projecto(carproj).espaco(k).ganhos_int(8).area = get(handles.✓
defespaco_intelement8_area,'String');
projecto(carproj).espaco(k).ganhos_int(8).temp = get(handles.✓
defespaco_intelement8_temp,'String');

clearvars x1 x2 y1 y2 y3

save projecto.mat projecto

end

save projecto.mat projecto
%%
warndlg('Espaço adicionado','Aviso !')
case 'Não'
case 'Cancelar'

end

% -----
function adespaco_adhorario_Callback(hObject, eventdata, handles)
% hObject    handle to adespaco_adhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_horario.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to adespaco_defhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to adespaco_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_espaco.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_adparede_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to adespaco_adparede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

close(findobj('type','figure','name','Adicionar Espaço'))

% -----
function adespaco_verparede_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_verparede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('ver_parede.m')

close(findobj('type','figure','name','Adicionar Espaço'))

function defespaco_hrint_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_hrint (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_hrint as text
%        str2double(get(hObject,'String')) returns contents of defespaco_hrint as a
double

% --- Executes during object creation, after setting all properties.
function defespaco_hrint_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_hrint (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_hnext_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_hnext (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_hnext as text
%        str2double(get(hObject,'String')) returns contents of defespaco_hnext as a
double

% --- Executes during object creation, after setting all properties.
function defespaco_hnext_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_hnext (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

```

```

%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function defespaco_caudalar_Callback(hObject, eventdata, handles)
% hObject      handle to defespaco_caudalar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of defespaco_caudalar as text
%          str2double(get(hObject,'String')) returns contents of defespaco_caudalar as
a double

% --- Executes during object creation, after setting all properties.
function defespaco_caudalar_CreateFcn(hObject, eventdata, handles)
% hObject      handle to defespaco_caudalar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function adespaco_elementos_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_elementos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function adespaco_espaco_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_espaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function adespaco_horario_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_horario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function adespaco_adespaco_Callback(hObject, eventdata, handles)
% hObject      handle to adespaco_adespaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```
% -----  
function adespaco_janela_Callback(hObject, eventdata, handles)  
% hObject    handle to adespaco_janela (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% -----  
function adespaco_porta_Callback(hObject, eventdata, handles)  
% hObject    handle to adespaco_porta (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% -----  
function adespaco_parede_Callback(hObject, eventdata, handles)  
% hObject    handle to adespaco_parede (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

```

function varargout = definir_horario(varargin)
% DEFINIR_HORARIO MATLAB code for definir_horario.fig
%     DEFINIR_HORARIO, by itself, creates a new DEFINIR_HORARIO or raises the
existing
%     singleton*.
%
%     H = DEFINIR_HORARIO returns the handle to a new DEFINIR_HORARIO or the handle
to
%     the existing singleton*.
%
%     DEFINIR_HORARIO('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in DEFINIR_HORARIO.M with the given input arguments.
%
%     DEFINIR_HORARIO('Property','Value',...) creates a new DEFINIR_HORARIO or
raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before definir_horario_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to definir_horario_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help definir_horario

% Last Modified by GUIDE v2.5 21-Jul-2016 11:28:55

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @definir_horario_OpeningFcn, ...
                  'gui_OutputFcn',  @definir_horario_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before definir_horario is made visible.
function definir_horario_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to definir_horario (see VARARGIN)

% Choose default command line output for definir_horario
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

movegui(gcf, 'center')

set(handles.figure1, 'Name', 'Adicionar Horário')

set(handles.defhorario_lista, 'String', '(Selecione o tipo de horário)')

% UIWAIT makes definir_horario wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = definir_horario_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function defhorario_nome_Callback(hObject, eventdata, handles)
% hObject handle to defhorario_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of defhorario_nome as text
% str2double(get(hObject, 'String')) returns contents of defhorario_nome as a
double

% --- Executes during object creation, after setting all properties.
function defhorario_nome_CreateFcn(hObject, eventdata, handles)
% hObject handle to defhorario_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiControlBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in horario_ocupacao.
function horario_ocupacao_Callback(hObject, eventdata, handles)
% hObject handle to horario_ocupacao (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject, 'Value') returns toggle state of horario_ocupacao

% --- Executes on button press in horario_equipamentos.

```

```

function horario_equipamentos_Callback(hObject, eventdata, handles)
% hObject      handle to horario_equipamentos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of horario_equipamentos

% --- Executes on button press in horario_iluminacao.
function horario_iluminacao_Callback(hObject, eventdata, handles)
% hObject      handle to horario_iluminacao (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of horario_iluminacao

% --- Executes on button press in defhorario_visualizar.
function defhorario_visualizar_Callback(hObject, eventdata, handles)
% hObject      handle to defhorario_visualizar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

load projecto.mat

global carproj

k = get(handles.defhorario_lista, 'Value');

switch get(handles.defhorario_tipo, 'Value')

    case 1
    case 2 % Ocupação
        set(handles.defhorario_nome, 'String', projecto(carproj).bdados.horario.✓
ocupacao(k).nome)
        set(handles.defhorario_horario, 'Data', projecto(carproj).bdados.horario.✓
ocupacao(k).data)
    case 3 % Iluminação
        set(handles.defhorario_nome, 'String', projecto(carproj).bdados.horario.✓
iluminacao(k).nome)
        set(handles.defhorario_horario, 'Data', projecto(carproj).bdados.horario.✓
iluminacao(k).data)
    case 4 % Equipamentos
        set(handles.defhorario_nome, 'String', projecto(carproj).bdados.horario.✓
equipamentos(k).nome)
        set(handles.defhorario_horario, 'Data', projecto(carproj).bdados.horario.✓
equipamentos(k).data)
    case 5 %
        set(handles.defhorario_nome, 'String', projecto(carproj).bdados.horario.✓
diversas(k).nome)
        set(handles.defhorario_horario, 'Data', projecto(carproj).bdados.horario.✓
diversas(k).data)
end

clearvars k

% --- Executes on button press in defhorario_guardar.
function defhorario_guardar_Callback(hObject, eventdata, handles)
% hObject      handle to defhorario_guardar (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

choice = questdlg('Deseja adicionar o horário', 'Adicionar✓
Horário', 'Sim', 'Não', 'Cancelar', 'Cancelar');

load projecto.mat

global carproj

switch choice
    case 'Sim'

        switch get(handles.defhorario_tipo, 'Value')
            case 1
                warndlg('Escolha o tipo de horário', 'Erro !')
            case 2
                k = length(projecto(carproj).bdados.horario.ocupacao)+1;
                projecto(carproj).bdados.horario.ocupacao(k).nome = get(handles.✓
defhorario_nome, 'String');
                projecto(carproj).bdados.horario.ocupacao(k).data = get(handles.✓
defhorario_horario, 'Data');

                save projecto.mat projecto

                warndlg('Horário adicionado', 'Aviso!')
            case 3
                k = length(projecto(carproj).bdados.horario.iluminacao)+1;
                projecto(carproj).bdados.horario.iluminacao(k).nome = get(handles.✓
defhorario_nome, 'String');
                projecto(carproj).bdados.horario.iluminacao(k).data = get(handles.✓
defhorario_horario, 'Data');

                save projecto.mat projecto

                warndlg('Horário adicionado', 'Aviso!')
            case 4
                k = length(projecto(carproj).bdados.horario.equipamentos)+1;
                projecto(carproj).bdados.horario.equipamentos(k).nome = get(handles.✓
defhorario_nome, 'String');
                projecto(carproj).bdados.horario.equipamentos(k).data = get(handles.✓
defhorario_horario, 'Data');

                save projecto.mat projecto

                warndlg('Horário adicionado', 'Aviso!')
            case 5
                k = length(projecto(carproj).bdados.horario.diversas)+1;
                projecto(carproj).bdados.horario.diversas(k).nome = get(handles.✓
defhorario_nome, 'String');
                projecto(carproj).bdados.horario.diversas(k).data = get(handles.✓
defhorario_horario, 'Data');

                save projecto.mat projecto

                warndlg('Horário adicionado', 'Aviso!')
        end
    end
end

```



```

        case 'Não'

        case 'Cancelar'
    end

clearvars choice k

% --- Executes on button press in defhorario_eliminar.
function defhorario_eliminar_Callback(hObject, eventdata, handles)
% hObject      handle to defhorario_eliminar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

k = get(handles.defhorario_lista, 'value');

        choice = questdlg('Deseja eliminar o horário?', 'Eliminar↵
Horário', 'Sim', 'Não', 'Cancelar', 'Cancelar');

        switch choice
            case 'Sim'

                load projecto.mat

                global carproj

                if k>15
                    switch get(handles.defhorario_tipo, 'Value')

                        case 1
                        case 2

                            projecto(carproj).bdados.horario.ocupacao(k)=[];
                            case 3
                            projecto(carproj).bdados.horario.iluminacao(k)=[];
                            case 4
                            projecto(carproj).bdados.horario.equipamentos(k)=[];
                            case 5
                            projecto(carproj).bdados.horario.diversas(k)=[];
                            end

                            set(handles.defhorario_lista, 'Value', 1)

                            save projecto.mat projecto

                            warndlg('Horário eliminado', 'Aviso !')
                    elseif get(handles.defhorario_tipo, 'Value')==5

                        projecto(carproj).bdados.horario.diversas(k)=[];
                        save projecto.mat projecto

                        warndlg('Horário eliminado', 'Aviso !')
                else
                    warndlg('Não é possível eliminar este horário', 'Erro !')
                end
            case 'Não'
            case 'Cancelar'

```

```

end
set(handles.defhorario_lista,'value',1)
set(handles.defhorario_nome,'string','')
set(handles.defhorario_tipo,'value',1)
set(handles.defhorario_horario, 'Data', []);

clearvars choice k

% --- Executes on selection change in defhorario_lista.
function defhorario_lista_Callback(hObject, eventdata, handles)
% hObject    handle to defhorario_lista (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defhorario_lista contents✓
as cell array
%         contents{get(hObject,'Value')} returns selected item from defhorario_lista

% --- Executes during object creation, after setting all properties.
function defhorario_lista_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defhorario_lista (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on defhorario_visualizar and none of its✓
controls.
function defhorario_visualizar_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to defhorario_visualizar (see GCBO)
% eventdata  structure with the following fields (see UICONTROL)
%   Key: name of the key that was pressed, in lower case
%   Character: character interpretation of the key(s) that was pressed
%   Modifier: name(s) of the modifier key(s) (i.e., control, shift) pressed
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function defhorario_guardar_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defhorario_guardar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes on selection change in defhorario_tipo.
function defhorario_tipo_Callback(hObject, eventdata, handles)
% hObject    handle to defhorario_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns defhorario_tipo contents✓
as cell array
%         contents{get(hObject,'Value')} returns selected item from defhorario_tipo

```

```

load projecto.mat

global carproj

switch get(handles.defhorario_tipo,'Value')
    case 1
        set(handles.defhorario_lista,'String','(Selecione o tipo de horário)')
    case 2
        set(handles.defhorario_lista,'String',{projecto(carproj).bdados.horario.✓
ocupacao.nome})
    case 3
        set(handles.defhorario_lista,'String',{projecto(carproj).bdados.horario.✓
iluminacao.nome})
    case 4
        set(handles.defhorario_lista,'String',{projecto(carproj).bdados.horario.✓
equipamentos.nome})
    case 5
        if isempty(projecto(carproj).bdados.horario.diversas)==1
            set(handles.defhorario_lista,'String','(Não existem horários)')
        else
            set(handles.defhorario_lista,'String',{projecto(carproj).bdados.horario.✓
diversas.nome})
        end
end

set(handles.defhorario_lista,'Value',1)

% --- Executes during object creation, after setting all properties.
function defhorario_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to defhorario_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in defhorario_carregar.
function defhorario_carregar_Callback(hObject, eventdata, handles)
% hObject    handle to defhorario_carregar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

filename = uigetfile('.xlsx');

a=xlsread(filename);

set(handles.defhorario_horario,'Data',a)

% -----
function adhorario_inicio_Callback(hObject, eventdata, handles)
% hObject    handle to adhorario_inicio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('inicio.m')

```

```

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_local_Callback(hObject, eventdata, handles)
% hObject    handle to adhorario_local (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('local.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_materiais_Callback(hObject, eventdata, handles)
% hObject    handle to adhorario_materiais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('registamateriais.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_relatorios_Callback(hObject, eventdata, handles)
% hObject    handle to adhorario_relatorios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to adhorario_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('vidro.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_caixilho_Callback(hObject, eventdata, handles)
% hObject    handle to adhorario_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('caixilho.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_adporta_Callback(hObject, eventdata, handles)
% hObject    handle to adhorario_adporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('Porta.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_verporta_Callback(hObject, eventdata, handles)
% hObject    handle to adhorario_verporta (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('porta_vereliminar.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_simples_Callback(hObject, eventdata, handles)
% hObject handle to adhorario_simples (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('janela_simples.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_dupla_Callback(hObject, eventdata, handles)
% hObject handle to adhorario_dupla (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('janela_dupla.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_duplafolha_Callback(hObject, eventdata, handles)
% hObject handle to adhorario_duplafolha (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_verjanela_Callback(hObject, eventdata, handles)
% hObject handle to adhorario_verjanela (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_defhorario_Callback(hObject, eventdata, handles)
% hObject handle to adhorario_defhorario (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_defespaco_Callback(hObject, eventdata, handles)
% hObject handle to adhorario_defespaco (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

close(findobj('type','figure','name','Adicionar Horário'))

```

```

% -----
function adhorario_verespaco_Callback(hObject, eventdata, handles)
% hObject      handle to adhorario_verespaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('ver_espaco.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_adparede_Callback(hObject, eventdata, handles)
% hObject      handle to adhorario_adparede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

close(findobj('type','figure','name','Adicionar Horário'))

% -----
function adhorario_verparede_Callback(hObject, eventdata, handles)
% hObject      handle to adhorario_verparede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('ver_parede.m')

close(findobj('type','figure','name','Adicionar Horário'))

% --- Executes on button press in defhorario_limpartabela.
function defhorario_limpartabela_Callback(hObject, eventdata, handles)
% hObject      handle to defhorario_limpartabela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

set(handles.defhorario_horario,'Data',[])

% -----
function adhorario_elementos_Callback(hObject, eventdata, handles)
% hObject      handle to adhorario_elementos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function adhorario_espaco_Callback(hObject, eventdata, handles)
% hObject      handle to adhorario_espaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function adhorario_horario_Callback(hObject, eventdata, handles)
% hObject      handle to adhorario_horario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----

```

```
function adhorario_adhorario_Callback(hObject, eventdata, handles)
% hObject      handle to adhorario_adhorario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function adhorario_janela_Callback(hObject, eventdata, handles)
% hObject      handle to adhorario_janela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function adhorario_porta_Callback(hObject, eventdata, handles)
% hObject      handle to adhorario_porta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function adhorario_parede_Callback(hObject, eventdata, handles)
% hObject      handle to adhorario_parede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```

function varargout = horario(varargin)
% HORARIO MATLAB code for horario.fig
%     HORARIO, by itself, creates a new HORARIO or raises the existing
%     singleton*.
%
%     H = HORARIO returns the handle to a new HORARIO or the handle to
%     the existing singleton*.
%
%     HORARIO('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in HORARIO.M with the given input arguments.
%
%     HORARIO('Property','Value',...) creates a new HORARIO or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before horario_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to horario_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help horario

% Last Modified by GUIDE v2.5 21-Jul-2016 11:32:27

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @horario_OpeningFcn, ...
                  'gui_OutputFcn',  @horario_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before horario is made visible.
function horario_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to horario (see VARARGIN)

% Choose default command line output for horario
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```



```

movegui(gcf,'center')

set(handles.figure1,'Name','Definir Horário')

set(handles.horario_tabocupacao,'Data',↵
{' ','1','1','1','1','1','1','1','1','1','1','1','1','1'})
set(handles.horario_tabiluminacao,'Data',↵
{' ','1','1','1','1','1','1','1','1','1','1','1','1','1'})
set(handles.horario_tabeletricos,'Data',↵
{' ','1','1','1','1','1','1','1','1','1','1','1','1','1'})
set(handles.horario_tabdiversas,'Data',↵
{' ','1','1','1','1','1','1','1','1','1','1','1','1','1'})

load projecto.mat

global carproj

if isempty(projecto(carproj).espaco)==1

    set(handles.horario_espaco,'String','(Selecione)')
else
    set(handles.horario_espaco,'String',{'(Selecione)',projecto(carproj).espaco.↵
nome})
end

% Ocupação
set(handles.horario_ocup1,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
ocupacao.nome})
set(handles.horario_ocup2,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
ocupacao.nome})
set(handles.horario_ocup3,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
ocupacao.nome})
set(handles.horario_ocup4,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
ocupacao.nome})

% Iluminação
set(handles.horario_ilum1,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
iluminacao.nome})
set(handles.horario_ilum2,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
iluminacao.nome})
set(handles.horario_ilum3,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
iluminacao.nome})
set(handles.horario_ilum4,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
iluminacao.nome})

% Equipamentos
set(handles.horario_eqe1,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
equipamentos.nome})
set(handles.horario_eqe2,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
equipamentos.nome})
set(handles.horario_eqe3,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
equipamentos.nome})
set(handles.horario_eqe4,'String',{'(Selecione)',projecto(carproj).bdados.horario.↵
equipamentos.nome})

```

```

% Diversas

if isempty(projecto(carproj).bdados.horario.diversas) ==1

else
set(handles.horario_div1,'String',{'(Selecione)',projecto(carproj).bdados.horario.✓
diversas.nome})
set(handles.horario_div2,'String',{'(Selecione)',projecto(carproj).bdados.horario.✓
diversas.nome})
set(handles.horario_div3,'String',{'(Selecione)',projecto(carproj).bdados.horario.✓
diversas.nome})
set(handles.horario_div4,'String',{'(Selecione)',projecto(carproj).bdados.horario.✓
diversas.nome})
end

% UIWAIT makes horario wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = horario_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in horario_espaco.
function horario_espaco_Callback(hObject, eventdata, handles)
% hObject      handle to horario_espaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_espaco contents✓
as cell array
%           contents{get(hObject,'Value')} returns selected item from horario_espaco

% --- Executes during object creation, after setting all properties.
function horario_espaco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to horario_espaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_ocup1.
function horario_ocup1_Callback(hObject, eventdata, handles)
% hObject      handle to horario_ocup1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_ocup1 contents as cell array
%           contents{get(hObject,'Value')} returns selected item from horario_ocup1

% --- Executes during object creation, after setting all properties.
function horario_ocup1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to horario_ocup1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_ocup2.
function horario_ocup2_Callback(hObject, eventdata, handles)
% hObject      handle to horario_ocup2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_ocup2 contents as cell array
%           contents{get(hObject,'Value')} returns selected item from horario_ocup2

% --- Executes during object creation, after setting all properties.
function horario_ocup2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to horario_ocup2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_ocup3.
function horario_ocup3_Callback(hObject, eventdata, handles)
% hObject      handle to horario_ocup3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_ocup3 contents as cell array
%           contents{get(hObject,'Value')} returns selected item from horario_ocup3

% --- Executes during object creation, after setting all properties.
function horario_ocup3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to horario_ocup3 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_ocup4.
function horario_ocup4_Callback(hObject, eventdata, handles)
% hObject handle to horario_ocup4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_ocup4 contents as
cell array
% contents{get(hObject,'Value')} returns selected item from horario_ocup4

% --- Executes during object creation, after setting all properties.
function horario_ocup4_CreateFcn(hObject, eventdata, handles)
% hObject handle to horario_ocup4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in horario_guardar.
function horario_guardar_Callback(hObject, eventdata, handles)
% hObject handle to horario_guardar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

choice = questdlg('Deseja definir o horário?', 'Definir
Horário', 'Sim', 'Não', 'Cancelar', 'Cancelar');

switch choice
    case 'Sim'
        load projecto.mat

        global carproj

        k = get(handles.horario_espaco, 'Value')-1;

        %Ocupação

        projecto(carproj).espaco(k).horario.ocupacao.tabela =get(handles.
horario_tabocupacao, 'Data');
        projecto(carproj).espaco(k).horario.ocupacao.um=get(handles.

```

```

horario_ocup1, 'Value')-1;
    projecto(carproj).espaco(k).horario.ocupacao.dois=get(handles.✓
horario_ocup2, 'Value')-1;
    projecto(carproj).espaco(k).horario.ocupacao.tres=get(handles.✓
horario_ocup3, 'Value')-1;
    projecto(carproj).espaco(k).horario.ocupacao.quatro=get(handles.✓
horario_ocup4, 'Value')-1;

    % Iluminação

    projecto(carproj).espaco(k).horario.iluminacao.tabela =get(handles.✓
horario_tabiluminacao, 'Data');
    projecto(carproj).espaco(k).horario.iluminacao.um=get(handles.✓
horario_ilum1, 'Value')-1;
    projecto(carproj).espaco(k).horario.iluminacao.dois=get(handles.✓
horario_ilum2, 'Value')-1;
    projecto(carproj).espaco(k).horario.iluminacao.tres=get(handles.✓
horario_ilum3, 'Value')-1;
    projecto(carproj).espaco(k).horario.iluminacao.quatro=get(handles.✓
horario_ilum4, 'Value')-1;

    % Equipamentos

    projecto(carproj).espaco(k).horario.equipamentos.tabela =get(handles.✓
horario_tabeletricos, 'Data');
    projecto(carproj).espaco(k).horario.equipamentos.um=get(handles.✓
horario_eqe1, 'Value')-1;
    projecto(carproj).espaco(k).horario.equipamentos.dois=get(handles.✓
horario_eqe2, 'Value')-1;
    projecto(carproj).espaco(k).horario.equipamentos.tres=get(handles.✓
horario_eqe3, 'Value')-1;
    projecto(carproj).espaco(k).horario.equipamentos.quatro=get(handles.✓
horario_eqe4, 'Value')-1;

    % Cargas diversas

    projecto(carproj).espaco(k).horario.diversas.tabela =get(handles.✓
horario_tabdiversas, 'Data');
    projecto(carproj).espaco(k).horario.diversas.um=get(handles.✓
horario_div1, 'Value')-1;
    projecto(carproj).espaco(k).horario.diversas.dois=get(handles.✓
horario_div2, 'Value')-1;
    projecto(carproj).espaco(k).horario.diversas.tres=get(handles.✓
horario_div3, 'Value')-1;
    projecto(carproj).espaco(k).horario.diversas.quatro=get(handles.✓
horario_div4, 'Value')-1;

    save projecto.mat projecto
    warndlg('Horários definidos.', 'Aviso !')
    case 'Não'
    case 'Cancelar'
end

% --- Executes on selection change in horario_ilum1.
function horario_ilum1_Callback(hObject, eventdata, handles)
% hObject    handle to horario_ilum1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject,'String')) returns horario_ilum1 contents as✓
cell array
%         contents{get(hObject,'Value')} returns selected item from horario_ilum1

% --- Executes during object creation, after setting all properties.
function horario_ilum1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to horario_ilum1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_ilum2.
function horario_ilum2_Callback(hObject, eventdata, handles)
% hObject    handle to horario_ilum2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_ilum2 contents as✓
cell array
%         contents{get(hObject,'Value')} returns selected item from horario_ilum2

% --- Executes during object creation, after setting all properties.
function horario_ilum2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to horario_ilum2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_ilum3.
function horario_ilum3_Callback(hObject, eventdata, handles)
% hObject    handle to horario_ilum3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_ilum3 contents as✓
cell array
%         contents{get(hObject,'Value')} returns selected item from horario_ilum3

% --- Executes during object creation, after setting all properties.
function horario_ilum3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to horario_ilum3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_ilum4.
function horario_ilum4_Callback(hObject, eventdata, handles)
% hObject     handle to horario_ilum4 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_ilum4 contents as
cell array
%     contents{get(hObject,'Value')} returns selected item from horario_ilum4

% --- Executes during object creation, after setting all properties.
function horario_ilum4_CreateFcn(hObject, eventdata, handles)
% hObject     handle to horario_ilum4 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_div1.
function horario_div1_Callback(hObject, eventdata, handles)
% hObject     handle to horario_div1 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_div1 contents as
cell array
%     contents{get(hObject,'Value')} returns selected item from horario_div1

% --- Executes during object creation, after setting all properties.
function horario_div1_CreateFcn(hObject, eventdata, handles)
% hObject     handle to horario_div1 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_div2.

```

```

function horario_div2_Callback(hObject, eventdata, handles)
% hObject      handle to horario_div2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_div2 contents as a
cell array
%           contents{get(hObject,'Value')} returns selected item from horario_div2

% --- Executes during object creation, after setting all properties.
function horario_div2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to horario_div2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_div3.
function horario_div3_Callback(hObject, eventdata, handles)
% hObject      handle to horario_div3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_div3 contents as a
cell array
%           contents{get(hObject,'Value')} returns selected item from horario_div3

% --- Executes during object creation, after setting all properties.
function horario_div3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to horario_div3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_div4.
function horario_div4_Callback(hObject, eventdata, handles)
% hObject      handle to horario_div4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_div4 contents as a
cell array
%           contents{get(hObject,'Value')} returns selected item from horario_div4

```



```

% --- Executes during object creation, after setting all properties.
function horario_div4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to horario_div4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_eqe1.
function horario_eqe1_Callback(hObject, eventdata, handles)
% hObject    handle to horario_eqe1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_eqe1 contents✓
as cell array
%         contents{get(hObject,'Value')} returns selected item from horario_eqe1

% --- Executes during object creation, after setting all properties.
function horario_eqe1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to horario_eqe1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_eqe2.
function horario_eqe2_Callback(hObject, eventdata, handles)
% hObject    handle to horario_eqe2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_eqe2 contents✓
as cell array
%         contents{get(hObject,'Value')} returns selected item from horario_eqe2

% --- Executes during object creation, after setting all properties.
function horario_eqe2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to horario_eqe2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_eqe3.
function horario_eqe3_Callback(hObject, eventdata, handles)
% hObject    handle to horario_eqe3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_eqe3 contents✓
as cell array
%         contents{get(hObject,'Value')} returns selected item from horario_eqe3

% --- Executes during object creation, after setting all properties.
function horario_eqe3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to horario_eqe3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in horario_eqe4.
function horario_eqe4_Callback(hObject, eventdata, handles)
% hObject    handle to horario_eqe4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns horario_eqe4 contents✓
as cell array
%         contents{get(hObject,'Value')} returns selected item from horario_eqe4

% --- Executes during object creation, after setting all properties.
function horario_eqe4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to horario_eqe4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function horario_inicio_Callback(hObject, eventdata, handles)
% hObject    handle to horario_inicio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('inicio.m')

```

```

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_local_Callback(hObject, eventdata, handles)
% hObject    handle to horario_local (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('local.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_materiais_Callback(hObject, eventdata, handles)
% hObject    handle to horario_materiais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('registamateriais.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_relatorios_Callback(hObject, eventdata, handles)
% hObject    handle to horario_relatorios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_adhorario_Callback(hObject, eventdata, handles)
% hObject    handle to horario_adhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_horario.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_defespaco_Callback(hObject, eventdata, handles)
% hObject    handle to horario_defespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to horario_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_espaco.m')

close(findobj('type','figure','name','Definir Horário'))

% -----

```

```

function horario_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to horario_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('vidro.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_caixilho_Callback(hObject, eventdata, handles)
% hObject      handle to horario_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('caixilho.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_parede_Callback(hObject, eventdata, handles)
% hObject      handle to horario_parede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_verparede_Callback(hObject, eventdata, handles)
% hObject      handle to horario_verparede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('ver_parede.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_adporta_Callback(hObject, eventdata, handles)
% hObject      handle to horario_adporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('Porta.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_verporta_Callback(hObject, eventdata, handles)
% hObject      handle to horario_verporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('porta_vereliminar.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_simples_Callback(hObject, eventdata, handles)
% hObject      handle to horario_simples (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_simples.m')

```

```

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_dupla_Callback(hObject, eventdata, handles)
% hObject    handle to horario_dupla (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_dupla.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_duplafolha_Callback(hObject, eventdata, handles)
% hObject    handle to horario_duplafolha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_verjanela_Callback(hObject, eventdata, handles)
% hObject    handle to horario_verjanela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')

close(findobj('type','figure','name','Definir Horário'))

% -----
function horario_elementos_Callback(hObject, eventdata, handles)
% hObject    handle to horario_elementos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function horario_esp_Callback(hObject, eventdata, handles)
% hObject    handle to horario_esp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function horario_horario_Callback(hObject, eventdata, handles)
% hObject    handle to horario_horario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function horario_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to horario_defhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----

```

```
function horario_janela_Callback(hObject, eventdata, handles)
% hObject      handle to horario_janela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function horario_porta_Callback(hObject, eventdata, handles)
% hObject      handle to horario_porta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function horario_pa_Callback(hObject, eventdata, handles)
% hObject      handle to horario_pa (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```

function varargout = inicio(varargin)
% INICIO MATLAB code for inicio.fig
%     INICIO, by itself, creates a new INICIO or raises the existing
%     singleton*.
%
%     H = INICIO returns the handle to a new INICIO or the handle to
%     the existing singleton*.
%
%     INICIO('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in INICIO.M with the given input arguments.
%
%     INICIO('Property','Value',...) creates a new INICIO or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before inicio_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to inicio_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help inicio

% Last Modified by GUIDE v2.5 21-Jul-2016 04:55:29

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @inicio_OpeningFcn, ...
                  'gui_OutputFcn',  @inicio_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before inicio is made visible.
function inicio_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to inicio (see VARARGIN)

% Choose default command line output for inicio
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```

```

movegui(gcf,'center')

set(handles.figure1, 'Name', 'Aplicação para o Cálculo da Condução Térmica em Edifícios')

addpath('imagens','base_dados_originais','funcoes')

% Carregar logos do ISEL e da ADEM

axes(handles.inicio_axes1);
imshow(imread('logo_ISEL.jpg'));

axes(handles.inicio_axes2);
imshow(imread('ADEM.jpg'));

% Carrega os projectos na lista de pojectos, quando o programa inicia.

load projecto.mat

if length(projecto)>0
    set(handles.inicio_lista,'String',{projecto.nome})
else
    set(handles.inicio_lista,'String','(Não existem projectos)')
end

%Abrir e enviar os projectos para a lista de projectos

% UIWAIT makes inicio wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = inicio_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in inicio_lista.
function inicio_lista_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_lista (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns inicio_lista contents as

```



```

cell array
%         contents{get(hObject,'Value')} returns selected item from inicio_lista

% --- Executes during object creation, after setting all properties.
function inicio_lista_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inicio_lista (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function inicio_nome_projecto_Callback(hObject, eventdata, handles)
% hObject    handle to inicio_nome_projecto (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inicio_nome_projecto as text
%         str2double(get(hObject,'String')) returns contents of inicio_nome_projecto
as a double

% --- Executes during object creation, after setting all properties.
function inicio_nome_projecto_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inicio_nome_projecto (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function inicio_nome_instituicao_Callback(hObject, eventdata, handles)
% hObject    handle to inicio_nome_instituicao (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inicio_nome_instituicao as text
%         str2double(get(hObject,'String')) returns contents of
inicio_nome_instituicao as a double

% --- Executes during object creation, after setting all properties.
function inicio_nome_instituicao_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inicio_nome_instituicao (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function inicio_nome_autor_Callback(hObject, eventdata, handles)
% hObject     handle to inicio_nome_autor (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inicio_nome_autor as text
%     str2double(get(hObject,'String')) returns contents of inicio_nome_autor as
a double

% --- Executes during object creation, after setting all properties.
function inicio_nome_autor_CreateFcn(hObject, eventdata, handles)
% hObject     handle to inicio_nome_autor (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in inicio_novo_projecto.
function inicio_novo_projecto_Callback(hObject, eventdata, handles)
% hObject     handle to inicio_novo_projecto (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%Cria a variável global projecto
global projecto

load projecto.mat

%Conta o número de linhas da variável projecto e adiciona 2 a esse valor
%k é o índice, que define o numero de entradas

k = length(projecto)+1;

choice = questdlg('Tem a certeza que deseja adicionar o novo
projecto','Atenção!','Sim','Não','Cancelar','Cancelar');

switch choice
    case 'Sim'
        %Vai buscar os campos preenchidos do Gui.
        projecto(k).nome=get(handles.inicio_nome_projecto,'String');
        projecto(k).instituicao=get(handles.inicio_nome_instituicao,'String');
        projecto(k).autor=get(handles.inicio_nome_autor,'String');

        if isempty(projecto(k).nome)

```

```

        projecto(k).nome = 'Sem nome';
    end
    if isempty(projecto(k).instituicao)
        projecto(k).instituicao = 'Sem instituição';
    end
    if isempty(projecto(k).autor)
        projecto(k).autor='Sem autor';
    end

    addpath('base_dados_originais')

    load clima.mat
    load material.mat
    load vidros.mat
    load caixilho.mat
    load janelas.mat
    load porta.mat
    load horario.mat
    load elementos.mat

    save subprojecto.mat clima material vidro caixilho janela porta horario✓
    elementos

    projecto(k).bdados = load('subprojecto.mat');

    delete subprojecto.mat
    projecto(k).local = [];
    projecto(k).espaco = [];

    %Guardar a variável projecto
    save projecto.mat projecto

    % Limpa os campos a preencher
    set(handles.inicio_nome_projecto,'string','')
    set(handles.inicio_nome_instituicao,'string','')
    set(handles.inicio_nome_autor,'string','')
case 'Não'
    % Limpa os campos a preencher
    set(handles.inicio_nome_projecto,'string','')
    set(handles.inicio_nome_instituicao,'string','')
    set(handles.inicio_nome_autor,'string','')

case 'Cancelar'
    % Limpa os campos a preencher
    set(handles.inicio_nome_projecto,'string','')
    set(handles.inicio_nome_instituicao,'string','')
    set(handles.inicio_nome_autor,'string','')

end

if length(projecto) > 0
    %Envio os dados para a lista de projectos
    set(handles.inicio_lista,'string',{projecto(1:end).nome})
else
    set(handles.inicio_lista,'string',{'(Não existem projectos)'})
end
end

```

```
clearvars k choice
```

```
% --- Executes on button press in inicio_carregar_projecto.
```

```
function inicio_carregar_projecto_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to inicio_carregar_projecto (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
%Carregar o ficheiro
```

```
load('projecto.mat')
```

```
k = get(handles.inicio_lista, 'value');
```

```
global carproj %Variavel global , que serve para carregar a base de dados de  
materiais, horarios, etc do respectivo projecto.
```

```
carproj = k;
```

```
choice = questdlg('Tem a certeza que deseja carregar o  
projecto', 'Atenção!', 'Sim', 'Não', 'Cancelar', 'Cancelar');
```

```
switch choice
```

```
    case 'Sim'
```

```
        if length(projecto)>0
```

```
            set(handles.inicio_nome_projecto, 'string', projecto(k).nome)
```

```
            set(handles.inicio_nome_instituicao, 'string', projecto(k).instituicao)
```

```
            set(handles.inicio_nome_autor, 'string', projecto(k).autor)
```

```
        else
```

```
            errordlg('Não existem projectos', 'Erro', 'modal')
```

```
        end
```

```
    case 'Não'
```

```
        set(handles.inicio_nome_projecto, 'string', '')
```

```
        set(handles.inicio_nome_instituicao, 'string', '')
```

```
        set(handles.inicio_nome_autor, 'string', '')
```

```
    case 'Cancelar'
```

```
        set(handles.inicio_nome_projecto, 'string', '')
```

```
        set(handles.inicio_nome_instituicao, 'string', '')
```

```
        set(handles.inicio_nome_autor, 'string', '')
```

```
end
```

```
clearvars k
```

```
% --- Executes on button press in inicio_eliminar_projecto.
```

```
function inicio_eliminar_projecto_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to inicio_eliminar_projecto (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
load('projecto.mat')
```

```
k = get(handles.inicio_lista, 'value');
```

```
    if length(projecto)>0
```

```
        choice = questdlg('Tem a certeza que deseja eliminar o  
projecto', 'Atenção!', 'Sim', 'Não', 'Cancelar', 'Cancelar');
```

```

switch choice
case 'Sim'

    projecto(k)= [];

    save projecto.mat projecto

    set(handles.inicio_nome_projecto,'string','')
    set(handles.inicio_nome_instituicao,'string','')
    set(handles.inicio_nome_autor,'string','')
case 'Não'
case 'Cancelar'
end

else
    errordlg('Não existem projectos','Erro','modal')

end

if length(projecto)>0
    set(handles.inicio_lista,'string',{projecto(1:end).nome})
    set(handles.inicio_lista,'value',1)
else
    set(handles.inicio_lista,'string','(Não existem projectos)')
    set(handles.inicio_lista,'value',1)
end

clearvars k

% --- Executes on button press in inicio_importar_projecto.
function inicio_importar_projecto_Callback(hObject, eventdata, handles)
% hObject    handle to inicio_importar_projecto (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global projecto
load projecto.mat

choice = questdlg('Tem a certeza que deseja importar um✔  
projecto','Atenção!','Sim','Não','Cancelar','Cancelar');

switch choice
case 'Sim'
    addpath('import')
    uiopen('*.mat')

    k = length(projecto)+1;

    projecto(k) = var_expor(1);

    save projecto.mat projecto

```

```

        case 'Não'
        case 'Cancelar'
end

if length(projecto)>0
    set(handles.inicio_lista,'String',{projecto(1:end).nome})
else
    set(handles.inicio_lista,'String',{'(Não existem projectos)'})
end

clearvars k

% --- Executes on button press in inicio_exportar_projecto.
function inicio_exportar_projecto_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_exportar_projecto (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
load projecto.mat

if length(projecto)>0

choice = questdlg('Tem a certeza que deseja exportar o ↙
projecto','Atenção!','Sim','Não','Cancelar','Cancelar');

switch choice

    case 'Sim'

        k = get(handles.inicio_lista,'value');

        var_expor= projecto(k);

        uisave({'var_expor'}, '')

    case 'Não'
    case 'Cancelar'
end
else
    errordlg('Não existem projectos','Erro','modal')
end
clearvars k var_expor

% --- Executes when figure1 is resized.
function figure1_ResizeFcn(hObject, eventdata, handles)
% hObject      handle to figure1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function inicio_inicio_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_inicio (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% -----
function inicio_local_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_local (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global carproj

if carproj >=1

run('local.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')

end

% -----
function inicio_elementos_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_elementos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function inicio_horarios_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_horarios (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function inicio_relatorios_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_relatorios (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('relatorios.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')

end

```

```

% -----
function inicio_materiais_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_materiais (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('registamateriais.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')
end

% -----
function vidro_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to vidro_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('vidro.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')
end

% -----
function inicio_caixilho_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

```



```

run('caixilho.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')

end

% -----
function inicio_adporta_Callback(hObject, eventdata, handles)
% hObject    handle to inicio_adporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('Porta.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')

end

% -----
function inicio_verporta_Callback(hObject, eventdata, handles)
% hObject    handle to inicio_verporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('porta_vereliminar.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')

end

% -----
function inicio_jansimples_Callback(hObject, eventdata, handles)
% hObject    handle to inicio_jansimples (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global carproj

```

```

if carproj >=1

run('janela_simples.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')
end

% -----
function inicio_jandupla_Callback(hObject, eventdata, handles)
% hObject    handle to inicio_jandupla (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global carproj

if carproj >=1

run('janela_dupla.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')
end

% -----
function inicio_duplafolha_Callback(hObject, eventdata, handles)
% hObject    handle to inicio_duplafolha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))
else
    errordlg('Projeto não foi carregado','Erro')
end

% -----
function inicio_verjanela_Callback(hObject, eventdata, handles)
% hObject    handle to inicio_verjanela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

global carproj

if carproj >=1

run('janela_vereliminar.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')
end

% -----
function inicio_adhorario_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_adhorario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('definir_horario.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')
end

% -----
function inicio_defhorario_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_defhorario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('horario.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))
else
    errordlg('Projeto não foi carregado','Erro')
end

% -----

```

```

function inicio_defespaco_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_defespaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global carproj

if carproj >=1

run('definir_espaco.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')

end

% -----
function inicio_verespaco_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_verespaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('ver_espaco.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')

end

% -----
function inicio_adparede_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_adparede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('adicionarparede.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))
else
    errordlg('Projeto não foi carregado','Erro')

end

```

```

% -----
function inicio_verparede_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_verparede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global carproj

if carproj >=1

run('ver_parede.m')

close(findobj('type','figure','name','Aplicação para o Cálculo da Condução Térmica✓
em Edifícios'))

else
    errordlg('Projeto não foi carregado','Erro')
end

% --- Executes during object creation, after setting all properties.
function inicio_novo_projecto_CreateFcn(hObject, eventdata, handles)
% hObject      handle to inicio_novo_projecto (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% -----

function inicio_espaco_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_espaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----

function inicio_janela_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_janela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----

function inicio_porta_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_porta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----

function inicio_parede_Callback(hObject, eventdata, handles)
% hObject      handle to inicio_parede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

function varargout = janela_dupla(varargin)
% JANELA_DUPLA MATLAB code for janela_dupla.fig
%     JANELA_DUPLA, by itself, creates a new JANELA_DUPLA or raises the existing
%     singleton*.
%
%     H = JANELA_DUPLA returns the handle to a new JANELA_DUPLA or the handle to
%     the existing singleton*.
%
%     JANELA_DUPLA('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in JANELA_DUPLA.M with the given input arguments.
%
%     JANELA_DUPLA('Property','Value',...) creates a new JANELA_DUPLA or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before janela_dupla_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to janela_dupla_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help janela_dupla

% Last Modified by GUIDE v2.5 21-Jul-2016 11:39:24

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @janela_dupla_OpeningFcn, ...
                  'gui_OutputFcn',  @janela_dupla_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before janela_dupla is made visible.
function janela_dupla_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to janela_dupla (see VARARGIN)

% Choose default command line output for janela_dupla
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```

```

movegui(gcf,'center')

set(handles.figure1,'Name','Adicionar Janela Dupla')

load projecto.mat

global carproj
projecto(carproj).bdados.janelasimples=[];

k = length(projecto(carproj).bdados.janela);

for x = 1:k

    s1 = 'Janela simples';
    y=strcmp(projecto(carproj).bdados.janela(x).tipo,s1);

    if y == 1

        z = length(projecto(carproj).bdados.janelasimples)+1;

        projecto(carproj).bdados.janelasimples(z).nome = projecto(carproj).bdados.✓
janela(x).nome;
        projecto(carproj).bdados.janelasimples(z).tipo = projecto(carproj).bdados.✓
janela(x).tipo;
        projecto(carproj).bdados.janelasimples(z).uw = projecto(carproj).bdados.✓
janela(x).uw;
    else
    end

end

save projecto.mat projecto

load projecto.mat

set(handles.jandupla_jansimplesint,'String',{'(Selecione)',projecto(carproj).bdados.✓
janelasimples.nome})
set(handles.jandupla_jansimplesext,'String',{'(Selecione)',projecto(carproj).bdados.✓
janelasimples.nome})

% UIWAIT makes janela_dupla wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = janela_dupla_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function jandupla_nome_Callback(hObject, eventdata, handles)
% hObject handle to jandupla_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jandupla_nome as text
%         str2double(get(hObject,'String')) returns contents of jandupla_nome as a double

% --- Executes during object creation, after setting all properties.
function jandupla_nome_CreateFcn(hObject, eventdata, handles)
% hObject      handle to jandupla_nome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in jandupla_imagem_areas.
function jandupla_imagem_areas_Callback(hObject, eventdata, handles)
% hObject      handle to jandupla_imagem_areas (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in jandupla_imagem_janelas.
function jandupla_imagem_janelas_Callback(hObject, eventdata, handles)
% hObject      handle to jandupla_imagem_janelas (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

function jandupla_coef_Callback(hObject, eventdata, handles)
% hObject      handle to jandupla_coef (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jandupla_coef as text
%         str2double(get(hObject,'String')) returns contents of jandupla_coef as a double

% --- Executes during object creation, after setting all properties.
function jandupla_coef_CreateFcn(hObject, eventdata, handles)
% hObject      handle to jandupla_coef (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



```

function jandupla_rsi_Callback(hObject, eventdata, handles)
% hObject      handle to jandupla_rsi (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jandupla_rsi as text
%         str2double(get(hObject,'String')) returns contents of jandupla_rsi as a
double

```

```

% --- Executes during object creation, after setting all properties.
function jandupla_rsi_CreateFcn(hObject, eventdata, handles)
% hObject      handle to jandupla_rsi (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function jandupla_rse_Callback(hObject, eventdata, handles)
% hObject      handle to jandupla_rse (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jandupla_rse as text
%         str2double(get(hObject,'String')) returns contents of jandupla_rse as a
double

```

```

% --- Executes during object creation, after setting all properties.
function jandupla_rse_CreateFcn(hObject, eventdata, handles)
% hObject      handle to jandupla_rse (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function jandupla_rs_Callback(hObject, eventdata, handles)
% hObject      handle to jandupla_rs (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jandupla_rs as text
%         str2double(get(hObject,'String')) returns contents of jandupla_rs as a
double

```

```

% --- Executes during object creation, after setting all properties.
function jandupla_rs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to jandupla_rs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in jandupla_guardar.
function jandupla_guardar_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_guardar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

choice = questdlg('Deseja adicionar a janela','', 'Sim', 'Não', 'Cancelar', 'Cancelar');

switch choice

    case 'Sim'

        load projecto.mat

        global carproj

        k = length(projecto(carproj).bdados.janela)+1;

        if get(handles.jandupla_checkuvalor, 'Value')==1

            projecto(carproj).bdados.janela(k).nome = get(handles.
jandupla_nome, 'String');
            projecto(carproj).bdados.janela(k).tipo = 'Janela dupla';
            projecto(carproj).bdados.janela(k).uw = str2double(get(handles.
jandupla_coef, 'String'));

        else

            y1 = get(handles.jandupla_jansimplesint, 'Value');
            x1 = projecto(carproj).bdados.janelasimples(y1-1).uw;
            y2 = get(handles.jandupla_jansimplesext, 'Value');
            x2 = projecto(carproj).bdados.janelasimples(y2-1).uw;
            x3 = str2double(get(handles.jandupla_rsi, 'String'));
            x4 = str2double(get(handles.jandupla_rse, 'String'));
            x5 = str2double(get(handles.jandupla_rs, 'String'));

            uw = 1/((1/x1)-x3+x5-x4+(1/x2));

            projecto(carproj).bdados.janela(k).nome = get(handles.
jandupla_nome, 'String');
            projecto(carproj).bdados.janela(k).tipo = 'Janela dupla';
            projecto(carproj).bdados.janela(k).uw = uw;

        end

        save projecto.mat projecto

```

```

        warndlg('Janel adicionada','Aviso !')
    case 'Não'
    case 'Cancelar'
end

clearvars k x1 x2 x3 x4 x5 y1 y2 uw choice

% -----
% hObject      handle to jandupla_inicio (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% -----
function jandupla_local_Callback(hObject, eventdata, handles)
% hObject      handle to jandupla_local (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)
run('local.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_materiais_Callback(hObject, eventdata, handles)
% hObject      handle to jandupla_materiais (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

run('registamateriais.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function dupla_relatorios_Callback(hObject, eventdata, handles)
% hObject      handle to dupla_relatorios (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to jandupla_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

run('vidro.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_caixilho_Callback(hObject, eventdata, handles)
% hObject      handle to jandupla_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

run('caixilho.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

```

```

% -----
function jandupla_adporta_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_adporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('Porta.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_verporta_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_verporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('porta_vereliminar.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_simples_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_simples (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('janela_simples.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_duplafolha_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_duplafolha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_verjanela_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_verjanela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_adhorario_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_adhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_horario.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_defhorario (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_defespaco_Callback(hObject, eventdata, handles)
% hObject handle to jandupla_defespaco (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_verespaco_Callback(hObject, eventdata, handles)
% hObject handle to jandupla_verespaco (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('ver_espaco.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_adparede_Callback(hObject, eventdata, handles)
% hObject handle to jandupla_adparede (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_verparede_Callback(hObject, eventdata, handles)
% hObject handle to jandupla_verparede (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('ver_parede.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% -----
function jandupla_inicio_Callback(hObject, eventdata, handles)
% hObject handle to jandupla_inicio (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

run('inicio.m')

close(findobj('type','figure','name','Adicionar Janela Dupla'))

% --- Executes on selection change in jandupla_jansimplesext.
function jandupla_jansimplesext_Callback(hObject, eventdata, handles)
% hObject handle to jandupla_jansimplesext (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject,'String')) returns jandupla_jansimplesext✓
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from✓
jandupla_jansimplesext

% --- Executes during object creation, after setting all properties.
function jandupla_jansimplesext_CreateFcn(hObject, eventdata, handles)
% hObject    handle to jandupla_jansimplesext (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in jandupla_jansimplesint.
function jandupla_jansimplesint_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_jansimplesint (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns jandupla_jansimplesint✓
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from✓
jandupla_jansimplesint

% --- Executes during object creation, after setting all properties.
function jandupla_jansimplesint_CreateFcn(hObject, eventdata, handles)
% hObject    handle to jandupla_jansimplesint (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in jandupla_checkuvalor.
function jandupla_checkuvalor_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_checkuvalor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of jandupla_checkuvalor

if get(handles.jandupla_checkuvalor,'Value')==1
    set(handles.jandupla_text334,'Visible','On')
    set(handles.jandupla_coef,'Visible','On')
    set(handles.jandupla_text335,'Visible','On')
else

```

```
    set(handles.jandupla_text334,'Visible','Off')
    set(handles.jandupla_coef,'Visible','Off')
    set(handles.jandupla_text335,'Visible','Off')
end
```

```
% -----
function jandupla_elementos_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_elementos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function jandupla_espaco_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_espaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function jandupla_horarios_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_horarios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function jandupla_janela_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_janela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function jandupla_porta_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_porta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function jandupla_parede_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_parede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function jandupla_jandupla_Callback(hObject, eventdata, handles)
% hObject    handle to jandupla_jandupla (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```

function varargout = janela_simple(s(varargin)
% JANELA_SIMPLES MATLAB code for janela_simple.fig
%     JANELA_SIMPLES, by itself, creates a new JANELA_SIMPLES or raises the
existing
%     singleton*.
%
%     H = JANELA_SIMPLES returns the handle to a new JANELA_SIMPLES or the handle
to
%     the existing singleton*.
%
%     JANELA_SIMPLES('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in JANELA_SIMPLES.M with the given input arguments.
%
%     JANELA_SIMPLES('Property','Value',...) creates a new JANELA_SIMPLES or raises
the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before janela_simple_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to janela_simple_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help janela_simple

% Last Modified by GUIDE v2.5 21-Jul-2016 05:24:09

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @janela_simple_OpeningFcn, ...
                  'gui_OutputFcn',  @janela_simple_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before janela_simple is made visible.
function janela_simple_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to janela_simple (see VARARGIN)

% Choose default command line output for janela_simple
handles.output = hObject;

```



```

% Update handles structure
guidata(hObject, handles);

movegui(gcf, 'center')

set(handles.figure1, 'Name', 'Adicionar Janela Simples')

load projecto.mat

global carproj

set(handles.jansimples_vidro, 'String', {'(Selecione)', projecto(carproj).bdados.vidro.✓
nome})

set(handles.jansimples_caixilho, 'String', {'(Selecione)', projecto(carproj).bdados.✓
caixilho.nome})

% UIWAIT makes janela_simples wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = janela_simples_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in jansimples_imagem_areas.
function jansimples_imagem_areas_Callback(hObject, eventdata, handles)
% hObject handle to jansimples_imagem_areas (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in jansimples_imagem_janelas.
function jansimples_imagem_janelas_Callback(hObject, eventdata, handles)
% hObject handle to jansimples_imagem_janelas (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in jansimples_guardar.
function jansimples_guardar_Callback(hObject, eventdata, handles)
% hObject handle to jansimples_guardar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

choice=questdlg('Deseja adicionar a janela','','Sim','Não','Cancelar','Cancelar');

switch choice

```

```

case 'Sim'
    load projecto.mat
    global carproj

    k = length(projecto(carproj).bdados.janela)+1;

    if get(handles.jansimples_check_ujanela, 'Value') == 1
        projecto(carproj).bdados.janela(k).nome=get(handles.✓
nome_janelasimp, 'String');
        projecto(carproj).bdados.janela(k).tipo='Janela simples';
        projecto(carproj).bdados.janela(k).uw=str2double(get(handles.✓
jansimples_coef_janela, 'String'));
    else
        y1 = get(handles.jansimples_vidro, 'Value');
        x1 = str2double(projecto(carproj).bdados.vidro(y1-1).ug);
        x2 = str2double(get(handles.jansimples_area_vidro, 'String'));
        y3 = get(handles.jansimples_caixilho, 'Value');
        x3 = str2double(projecto(carproj).bdados.caixilho(y3-1).uf);
        x4 = str2double(get(handles.jansimples_area_caixilho, 'String'));
        x5 = str2double(get(handles.jansimples_perimetro_vidro, 'String'));
        x6 = str2double(get(handles.jansimples_coef_linearvidro, 'String'));
        %Painel Opaco
        if get(handles.jansimples_check_opaco, 'Value')==1

            x7=str2double(get(handles.✓
jansimples_coef_opaco, 'String'));
            x8=str2double(get(handles.✓
jansimples_area_opaco, 'String'));
            x9=str2double(get(handles.✓
jansimples_perimetro_opaco, 'String'));
            x10=str2double(get(handles.✓
jansimples_coef_linearopaco, 'String'));
        else
            x7=0;
            x8=0;
            x9=0;
            x10=0;

        end
        % Dispositivo de Oclusão
        if get(handles.jansimples_check_oclusao, 'Value')==1
            x11 = str2double(get(handles.✓
jansimples_resist_oclusao, 'String'));
        else
            x11=0;
        end

        uw = ((x1*x2)+(x3*x4)+(x7*x8)+(x5*x6)+(x9*x10))/(x2+x4+x8);

        uws = 1/((1/uw)+x11);

        projecto(carproj).bdados.janela(k).nome = get(handles.✓
nome_janelasimp, 'String');
        projecto(carproj).bdados.janela(k).tipo = 'Janela simples';
        projecto(carproj).bdados.janela(k).uw = uws;
    end

    save projecto.mat projecto

```

```

        warndlg('Janela adicionada','Aviso !')
    case 'Não'
    case 'Cancelar'
end

clearvars k choice x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 y1 y3 uw uws

% --- Executes on button press in jansimples_check_ujanela.
function jansimples_check_ujanela_Callback(hObject, eventdata, handles)
% hObject      handle to jansimples_check_ujanela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of jansimples_check_ujanela

if get(handles.jansimples_check_ujanela,'Value')==1
    set(handles.panel_ujanela,'Visible','On')
else
    set(handles.panel_ujanela,'Visible','Off')
end

% --- Executes on button press in jansimples_check_oclusao.
function jansimples_check_oclusao_Callback(hObject, eventdata, handles)
% hObject      handle to jansimples_check_oclusao (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of jansimples_check_oclusao

if get(handles.jansimples_check_oclusao,'Value')==1

    set(handles.jansimples_panel_oclusao,'Visible','On')
else
    set(handles.jansimples_panel_oclusao,'Visible','Off')
end

% --- Executes on button press in jansimples_check_opaco.
function jansimples_check_opaco_Callback(hObject, eventdata, handles)
% hObject      handle to jansimples_check_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of jansimples_check_opaco

if get(handles.jansimples_check_opaco,'Value')==1

    set(handles.jansimples_painel_opaco,'Visible','On')
else
    set(handles.jansimples_painel_opaco,'Visible','Off')
end

function jansimples_resist_oclusao_Callback(hObject, eventdata, handles)
% hObject      handle to jansimples_resist_oclusao (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jansimples_resist_oclusao as text
%        str2double(get(hObject,'String')) returns contents of ↵

```

jansimples\_resist\_oclusao as a double

% --- Executes during object creation, after setting all properties.

```
function jansimples_resist_oclusao_CreateFcn(hObject, eventdata, handles)
% hObject    handle to jansimples_resist_oclusao (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function jansimples_coef_janela_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to jansimples_coef_janela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

% Hints: get(hObject,'String') returns contents of jansimples\_coef\_janela as text

% str2double(get(hObject,'String')) returns contents of

jansimples\_coef\_janela as a double

% --- Executes during object creation, after setting all properties.

```
function jansimples_coef_janela_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to jansimples_coef_janela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

% --- Executes on selection change in jansimples\_vidro.

```
function jansimples_vidro_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to jansimples_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

% Hints: contents = cellstr(get(hObject,'String')) returns jansimples\_vidro contents  
as cell array

% contents{get(hObject,'Value')} returns selected item from jansimples\_vidro

% --- Executes during object creation, after setting all properties.

```
function jansimples_vidro_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to jansimples_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

% Hint: popupmenu controls usually have a white background on Windows.

% See ISPC and COMPUTER.

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
```

```

        set(hObject,'BackgroundColor','white');
end

function jansimples_area_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to jansimples_area_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jansimples_area_vidro as text
%          str2double(get(hObject,'String')) returns contents of jansimples_area_vidro
as a double

% --- Executes during object creation, after setting all properties.
function jansimples_area_vidro_CreateFcn(hObject, eventdata, handles)
% hObject      handle to jansimples_area_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in jansimples_caixilho.
function jansimples_caixilho_Callback(hObject, eventdata, handles)
% hObject      handle to jansimples_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns jansimples_caixilho
contents as cell array
%          contents{get(hObject,'Value')} returns selected item from
jansimples_caixilho

% --- Executes during object creation, after setting all properties.
function jansimples_caixilho_CreateFcn(hObject, eventdata, handles)
% hObject      handle to jansimples_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function jansimples_area_caixilho_Callback(hObject, eventdata, handles)
% hObject      handle to jansimples_area_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jansimples_area_caixilho as text
%          str2double(get(hObject,'String')) returns contents of

```

```
jansimples_area_caixilho as a double
```

```
% --- Executes during object creation, after setting all properties.  
function jansimples_area_caixilho_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to jansimples_area_caixilho (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.  
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(✓  
(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function jansimples_perimetro_vidro_Callback(hObject, eventdata, handles)  
% hObject    handle to jansimples_perimetro_vidro (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of jansimples_perimetro_vidro as ✓  
text  
%         str2double(get(hObject,'String')) returns contents of ✓  
jansimples_perimetro_vidro as a double
```

```
% --- Executes during object creation, after setting all properties.  
function jansimples_perimetro_vidro_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to jansimples_perimetro_vidro (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.  
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(✓  
(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function jansimples_coef_linearvidro_Callback(hObject, eventdata, handles)  
% hObject    handle to jansimples_coef_linearvidro (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of jansimples_coef_linearvidro as ✓  
text  
%         str2double(get(hObject,'String')) returns contents of ✓  
jansimples_coef_linearvidro as a double
```

```
% --- Executes during object creation, after setting all properties.  
function jansimples_coef_linearvidro_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to jansimples_coef_linearvidro (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.  
%         See ISPC and COMPUTER.
```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function jansimples_coef_opaco_Callback(hObject, eventdata, handles)
% hObject      handle to jansimples_coef_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jansimples_coef_opaco as text
%          str2double(get(hObject,'String')) returns contents of jansimples_coef_opaco
as a double

% --- Executes during object creation, after setting all properties.
function jansimples_coef_opaco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to jansimples_coef_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function jansimples_area_opaco_Callback(hObject, eventdata, handles)
% hObject      handle to jansimples_area_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jansimples_area_opaco as text
%          str2double(get(hObject,'String')) returns contents of jansimples_area_opaco
as a double

% --- Executes during object creation, after setting all properties.
function jansimples_area_opaco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to jansimples_area_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function jansimples_perimetro_opaco_Callback(hObject, eventdata, handles)
% hObject      handle to jansimples_perimetro_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jansimples_perimetro_opaco as
text
%          str2double(get(hObject,'String')) returns contents of
jansimples_perimetro_opaco as a double

```

```

% --- Executes during object creation, after setting all properties.
function jansimples_perimetro_opaco_CreateFcn(hObject, eventdata, handles)
% hObject    handle to jansimples_perimetro_opaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function jansimples_coef_linearopaco_Callback(hObject, eventdata, handles)
% hObject    handle to jansimples_coef_linearopaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jansimples_coef_linearopaco as ✓
text
%         str2double(get(hObject,'String')) returns contents of ✓
jansimples_coef_linearopaco as a double

% --- Executes during object creation, after setting all properties.
function jansimples_coef_linearopaco_CreateFcn(hObject, eventdata, handles)
% hObject    handle to jansimples_coef_linearopaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nome_janelasimp_Callback(hObject, eventdata, handles)
% hObject    handle to nome_janelasimp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nome_janelasimp as text
%         str2double(get(hObject,'String')) returns contents of nome_janelasimp as a ✓
double

% --- Executes during object creation, after setting all properties.
function nome_janelasimp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nome_janelasimp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))

```



```

        set(hObject,'BackgroundColor','white');
end

% -----
function simples_inicio_Callback(hObject, eventdata, handles)
% hObject    handle to simples_inicio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('inicio.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_local_Callback(hObject, eventdata, handles)
% hObject    handle to simples_local (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('local.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_materiais_Callback(hObject, eventdata, handles)
% hObject    handle to simples_materiais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('registamateriais.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_relatorios_Callback(hObject, eventdata, handles)
% hObject    handle to simples_relatorios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to simples_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('vidro.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_caixilho_Callback(hObject, eventdata, handles)
% hObject    handle to simples_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

run('caixilho.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_adiporta_Callback(hObject, eventdata, handles)
% hObject      handle to simples_adiporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('Porta.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_verporta_Callback(hObject, eventdata, handles)
% hObject      handle to simples_verporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('porta_vereliminar.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_jandupla_Callback(hObject, eventdata, handles)
% hObject      handle to simples_jandupla (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_dupla.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_duplafolha_Callback(hObject, eventdata, handles)
% hObject      handle to simples_duplafolha (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_verjanela_Callback(hObject, eventdata, handles)
% hObject      handle to simples_verjanela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_adhorario_Callback(hObject, eventdata, handles)
% hObject      handle to simples_adhorario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('definir_horario.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

```

```

% -----
function simples_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to simples_defhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_defespaco_Callback(hObject, eventdata, handles)
% hObject    handle to simples_defespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to simples_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_espaco.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_adporta_Callback(hObject, eventdata, handles)
% hObject    handle to simples_adporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function simples_verparede_Callback(hObject, eventdata, handles)
% hObject    handle to simples_verparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('ver_parede.m')

close(findobj('type','figure','name','Adicionar Janela Simples'))

% -----
function jansimples_jansimples_Callback(hObject, eventdata, handles)
% hObject    handle to jansimples_jansimples (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function jansimples_elementos_Callback(hObject, eventdata, handles)
% hObject    handle to jansimples_elementos (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function jansimples_espaco_Callback(hObject, eventdata, handles)
% hObject handle to jansimples_espaco (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function jansimples_horario_Callback(hObject, eventdata, handles)
% hObject handle to jansimples_horario (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function jansimples_janela_Callback(hObject, eventdata, handles)
% hObject handle to jansimples_janela (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function jansimples_porta_Callback(hObject, eventdata, handles)
% hObject handle to jansimples_porta (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function jansimples_parede_Callback(hObject, eventdata, handles)
% hObject handle to jansimples_parede (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```

function varargout = janela_vereliminar(varargin)
% JANELA_VERELIMINAR MATLAB code for janela_vereliminar.fig
%     JANELA_VERELIMINAR, by itself, creates a new JANELA_VERELIMINAR or raises the
existing
%     singleton*.
%
%     H = JANELA_VERELIMINAR returns the handle to a new JANELA_VERELIMINAR or the
handle to
%     the existing singleton*.
%
%     JANELA_VERELIMINAR('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in JANELA_VERELIMINAR.M with the given input
arguments.
%
%     JANELA_VERELIMINAR('Property','Value',...) creates a new JANELA_VERELIMINAR
or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before janela_vereliminar_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to janela_vereliminar_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help janela_vereliminar

% Last Modified by GUIDE v2.5 21-Jul-2016 11:41:37

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @janela_vereliminar_OpeningFcn, ...
                  'gui_OutputFcn',  @janela_vereliminar_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before janela_vereliminar is made visible.
function janela_vereliminar_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to janela_vereliminar (see VARARGIN)

% Choose default command line output for janela_vereliminar
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

movegui(gcf, 'center')

set(handles.figure1, 'Name', 'Ver / Eliminar Janela')

load projecto.mat

global carproj

set(handles.verjanela_lista, 'String', {projecto(carproj).bdados.janela.nome})

% UIWAIT makes janela_vereliminar wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = janela_vereliminar_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in verjanela_eliminar.
function verjanela_eliminar_Callback(hObject, eventdata, handles)
% hObject handle to verjanela_eliminar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
if get(handles.verjanela_lista, 'Value') > 21

    choice = questdlg('Deseja eliminar a ↵
janela', '', 'Sim', 'Não', 'Cancelar', 'Cancelar');

    switch choice
        case 'Sim'
            load projecto.mat
            global carproj

            k = get(handles.verjanela_lista, 'Value');

            projecto(carproj).bdados.janela(k) = [];

            save projecto.mat projecto

            set(handles.verjanela_lista, 'Value', 1)
            set(handles.verjanela_lista, 'String', {projecto(carproj).bdados.janela.↵
nome})

            set(handles.verjanela_nome, 'String', '')
            set(handles.verjanela_tipo, 'String', '')
            set(handles.verjanela_coef, 'String', '')

```

```

        warndlg('Janela eliminada','Aviso !')

        case 'Não'
        case 'Cancelar'
    end
else
    errordlg('Não pode eliminar esta janela','Aviso !')
end

clearvars choice k

% --- Executes on selection change in verjanela_lista.
function verjanela_lista_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_lista (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns verjanela_lista contents ✓
as cell array
%         contents{get(hObject,'Value')} returns selected item from verjanela_lista

% --- Executes during object creation, after setting all properties.
function verjanela_lista_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verjanela_lista (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verjanela_coef_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_coef (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verjanela_coef as text
%         str2double(get(hObject,'String')) returns contents of verjanela_coef as a ✓
double

% --- Executes during object creation, after setting all properties.
function verjanela_coef_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verjanela_coef (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function verjanela_nome_Callback(hObject, eventdata, handles)
% hObject      handle to verjanela_nome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verjanela_nome as text
%          str2double(get(hObject,'String')) returns contents of verjanela_nome as a
double

% --- Executes during object creation, after setting all properties.
function verjanela_nome_CreateFcn(hObject, eventdata, handles)
% hObject      handle to verjanela_nome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in verjanela_ver.
function verjanela_ver_Callback(hObject, eventdata, handles)
% hObject      handle to verjanela_ver (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
load projecto.mat
global carproj

k=get(handles.verjanela_lista,'Value');

set(handles.verjanela_nome,'String',projecto(carproj).bdados.janela(k).nome)
set(handles.verjanela_tipo,'String',projecto(carproj).bdados.janela(k).tipo)
set(handles.verjanela_coef,'String',projecto(carproj).bdados.janela(k).uw)

clearvars k

% --- Executes during object creation, after setting all properties.
function verjanela_ver_CreateFcn(hObject, eventdata, handles)
% hObject      handle to verjanela_ver (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function verjanela_tipo_CreateFcn(hObject, eventdata, handles)
% hObject      handle to verjanela_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))

```



```

        set(hObject,'BackgroundColor','white');
end

function verjanela_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to verjanela_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verjanela_tipo as text
%          str2double(get(hObject,'String')) returns contents of verjanela_tipo as a double

% -----
function verjanela_inicio_Callback(hObject, eventdata, handles)
% hObject      handle to verjanela_inicio (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('inicio.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_local_Callback(hObject, eventdata, handles)
% hObject      handle to verjanela_local (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('local.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_materiais_Callback(hObject, eventdata, handles)
% hObject      handle to verjanela_materiais (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('registamateriais.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_relatorios_Callback(hObject, eventdata, handles)
% hObject      handle to verjanela_relatorios (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to verjanela_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('vidro.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

```

```

% -----
function verjanela_caixilho_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('caixilho.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_adporta_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_adporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('Porta.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_verporta_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_verporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('porta_vereliminar.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_simples_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_simples (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_simples.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_dupla_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_dupla (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_dupla.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_duplafolha_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_duplafolha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_adhorario_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to verjanela_adhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_horario.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_defhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_defespaco_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_defespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('verespaco.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_adparede_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_adparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_verparede_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_verparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_parede.m')

close(findobj('type','figure','name','Ver / Eliminar Janela'))

% -----
function verjanela_elementos_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_elementos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% -----
function verjanela_espaco_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_espaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function verjanela_horario_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_horario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function verjanela_janela_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_janela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function verjanela_porta_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_porta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function verjanela_parede_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_parede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function verjanela_verjanela_Callback(hObject, eventdata, handles)
% hObject    handle to verjanela_verjanela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

function varargout = janela_duplafolhamesmoaro(varargin)
% JANELA_DUPLAFOLHAMESMOARO MATLAB code for janela_duplafolhamesmoaro.fig
%     JANELA_DUPLAFOLHAMESMOARO, by itself, creates a new JANELA_DUPLAFOLHAMESMOARO
or raises the existing
%     singleton*.
%
%     H = JANELA_DUPLAFOLHAMESMOARO returns the handle to a new
JANELA_DUPLAFOLHAMESMOARO or the handle to
%     the existing singleton*.
%
%     JANELA_DUPLAFOLHAMESMOARO('CALLBACK', hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in JANELA_DUPLAFOLHAMESMOARO.M with the given input
arguments.
%
%     JANELA_DUPLAFOLHAMESMOARO('Property','Value',...) creates a new
JANELA_DUPLAFOLHAMESMOARO or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before janela_duplafolhamesmoaro_OpeningFcn gets called.
An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to janela_duplafolhamesmoaro_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help janela_duplafolhamesmoaro

% Last Modified by GUIDE v2.5 21-Jul-2016 11:40:16

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @janela_duplafolhamesmoaro_OpeningFcn, ...
                  'gui_OutputFcn',  @janela_duplafolhamesmoaro_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before janela_duplafolhamesmoaro is made visible.
function janela_duplafolhamesmoaro_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to janela_duplafolhamesmoaro (see VARARGIN)

```

```

% Choose default command line output for janela_duplafolhamesmoaro
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

movegui(gcf, 'center')

set(handles.figure1, 'Name', 'Adicionar Janela de Dupla Folha no Mesmo Aro')

load projecto.mat
global carproj

set(handles.duplafolha_vidro_interior, 'String', {'(Selecione)', projecto(carproj).bdados.vidro.nome})
set(handles.duplafolha_vidro_exterior, 'String', {'(Selecione)', projecto(carproj).bdados.vidro.nome})
set(handles.duplafolha_listacaixilho, 'String', {'(Selecione)', projecto(carproj).bdados.caixilho.nome})

% UIWAIT makes janela_duplafolhamesmoaro wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = janela_duplafolhamesmoaro_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function duplafolha_nome_Callback(hObject, eventdata, handles)
% hObject handle to duplafolha_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of duplafolha_nome as text
% str2double(get(hObject, 'String')) returns contents of duplafolha_nome as a double

% --- Executes during object creation, after setting all properties.
function duplafolha_nome_CreateFcn(hObject, eventdata, handles)
% hObject handle to duplafolha_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in duplafolha_imagem_areas.
function duplafolha_imagem_areas_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_imagem_areas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in duplafolha_imagem_janelas.
function duplafolha_imagem_janelas_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_imagem_janelas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in duplafolha_guardar.
function duplafolha_guardar_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_guardar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

choice = questdlg('Deseja adicionar a
janela?', '', 'Sim', 'Não', 'Cancelar', 'Cancelar');

switch choice
case 'Sim'
    load projecto.mat

    global carproj

    k = length(projecto(carproj).bdados.janela)+1;

    if get(handles.duplafolha_check_ujanela,'Value')==1

        projecto(carproj).bdados.janela(k).nome = get(handles.
duplafolha_nome,'String');
        projecto(carproj).bdados.janela(k).tipo = 'Janela de dupla folha no
mesmo aro';
        projecto(carproj).bdados.janela(k).uw = str2double(get(handles.
duplafolha_coef_janela,'String'));
    else

        % Cálculo de Ug- vidro combinado

        y1 = get(handles.duplafolha_vidro_interior,'Value');
        x1 = str2double(projecto(carproj).bdados.vidro(y1-1).ug);
        y2 = get(handles.duplafolha_vidro_exterior,'Value');
        x2 = str2double(projecto(carproj).bdados.vidro(y2-1).ug);
        x3 = str2double(get(handles.duplafolha_rsi,'String'));
        x4 = str2double(get(handles.duplafolha_rse,'String'));
        x5 = str2double(get(handles.duplafolha_rs,'String'));

        ug = 1/((1/x1)-x3+x5-x4+(1/x2));

        x6 = str2double(get(handles.duplafolha_area_vidro,'String'));
        y7 = get(handles.duplafolha_listacaixilho,'Value');
        x7 = str2double(projecto(carproj).bdados.caixilho(y7-1).uf);

```

```

x8 = str2double(get(handles.duplafolha_area_caixilho, 'String'));
x9 = str2double(get(handles.duplafolha_perimetro_vidro, 'String'));
x10= str2double(get(handles.duplafolha_linear_vidro, 'String'));
    %Painel Opaco
    if get(handles.duplafolha_check_opaco, 'Value')==1

        x11=str2double(get(handles.✓
duplafolha_uopaco, 'String'));
        x12=str2double(get(handles.✓
duplafolha_area_opaco, 'String'));
        x13=str2double(get(handles.✓
duplafolha_perimetro_opaco, 'String'));
        x14=str2double(get(handles.✓
duplafolha_linear_opaco, 'String'));
    else
        x11=0;
        x12=0;
        x13=0;
        x14=0;

    end
    % Dispositivo de Oclusão
    if get(handles.duplafolha_check_oclusao, 'Value')==1
        x15 = str2double(get(handles.✓
duplafolha_resist_oclusao, 'String'));
    else
        x15=0;
    end

    uw = ((x6*ug)+(x12*x11)+(x8*x7)+(x9*x10)+(x13*x14))/(x6+x12+x8);

    uws = 1/((1/uw)+x15);

    projecto(carproj).bdados.janela(k).nome = get(handles.✓
duplafolha_nome, 'String');
    projecto(carproj).bdados.janela(k).tipo = 'Janela de dupla folha no✓
mesmo aro';
    projecto(carproj).bdados.janela(k).uw = uws;

    end
    save projecto.mat projecto
    warndlg('Janela adicionada', 'Aviso !')

    case 'Não'
    case 'Cancelar'
end
save projecto.mat projecto
clearvars choice k x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 y1 y2 ug uw✓
uws

% --- Executes on button press in duplafolha_check_ujanela.
function duplafolha_check_ujanela_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_check_ujanela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject, 'Value') returns toggle state of duplafolha_check_ujanela
if get(handles.duplafolha_check_ujanela, 'Value')==1
    set(handles.duplafolha_panel_coef, 'Visible', 'On')
else

```



```

    set(handles.duplafolha_panel_coef, 'Visible', 'Off')
end

% --- Executes on button press in duplafolha_check_oclusao.
function duplafolha_check_oclusao_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_check_oclusao (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject, 'Value') returns toggle state of duplafolha_check_oclusao
if get(handles.duplafolha_check_oclusao, 'Value')==1
    set(handles.duplafolha_panel_oclusao, 'Visible', 'On')
else
    set(handles.duplafolha_panel_oclusao, 'Visible', 'Off')
end

% --- Executes on button press in duplafolha_check_opaco.
function duplafolha_check_opaco_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_check_opaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject, 'Value') returns toggle state of duplafolha_check_opaco

if get(handles.duplafolha_check_opaco, 'Value')==1
    set(handles.duplafolha_panel_opaco, 'Visible', 'On')
else
    set(handles.duplafolha_panel_opaco, 'Visible', 'Off')
end

function duplafolha_resist_oclusao_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_resist_oclusao (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of duplafolha_resist_oclusao as text
%        str2double(get(hObject, 'String')) returns contents of
duplafolha_resist_oclusao as a double

% --- Executes during object creation, after setting all properties.
function duplafolha_resist_oclusao_CreateFcn(hObject, eventdata, handles)
% hObject    handle to duplafolha_resist_oclusao (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function duplafolha_coef_janela_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_coef_janela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of duplafolha_coef_janela as text
%         str2double(get(hObject,'String')) returns contents of
duplafolha_coef_janela as a double

% --- Executes during object creation, after setting all properties.
function duplafolha_coef_janela_CreateFcn(hObject, eventdata, handles)
% hObject    handle to duplafolha_coef_janela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in duplafolha_vidro_interior.
function duplafolha_vidro_interior_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_vidro_interior (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns duplafolha_vidro_interior
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
duplafolha_vidro_interior

% --- Executes during object creation, after setting all properties.
function duplafolha_vidro_interior_CreateFcn(hObject, eventdata, handles)
% hObject    handle to duplafolha_vidro_interior (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function duplafolha_area_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_area_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of duplafolha_area_vidro as text
%         str2double(get(hObject,'String')) returns contents of duplafolha_area_vidro
as a double

% --- Executes during object creation, after setting all properties.
function duplafolha_area_vidro_CreateFcn(hObject, eventdata, handles)
% hObject    handle to duplafolha_area_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in duplafolha_listacaixilho.
function duplafolha_listacaixilho_Callback(hObject, eventdata, handles)
% hObject     handle to duplafolha_listacaixilho (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns duplafolha_listacaixilho
contents as cell array
%     contents{get(hObject,'Value')} returns selected item from
duplafolha_listacaixilho

% --- Executes during object creation, after setting all properties.
function duplafolha_listacaixilho_CreateFcn(hObject, eventdata, handles)
% hObject     handle to duplafolha_listacaixilho (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function duplafolha_area_caixilho_Callback(hObject, eventdata, handles)
% hObject     handle to duplafolha_area_caixilho (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of duplafolha_area_caixilho as text
%     str2double(get(hObject,'String')) returns contents of
duplafolha_area_caixilho as a double

% --- Executes during object creation, after setting all properties.
function duplafolha_area_caixilho_CreateFcn(hObject, eventdata, handles)
% hObject     handle to duplafolha_area_caixilho (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function duplafolha_perimetro_vidro_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to duplafolha_perimetro_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of duplafolha_perimetro_vidro as
text
%           str2double(get(hObject,'String')) returns contents of
duplafolha_perimetro_vidro as a double

% --- Executes during object creation, after setting all properties.
function duplafolha_perimetro_vidro_CreateFcn(hObject, eventdata, handles)
% hObject      handle to duplafolha_perimetro_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function duplafolha_linear_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_linear_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of duplafolha_linear_vidro as text
%           str2double(get(hObject,'String')) returns contents of
duplafolha_linear_vidro as a double

% --- Executes during object creation, after setting all properties.
function duplafolha_linear_vidro_CreateFcn(hObject, eventdata, handles)
% hObject      handle to duplafolha_linear_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in duplafolha_vidro_exterior.
function duplafolha_vidro_exterior_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_vidro_exterior (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns duplafolha_vidro_exterior
contents as cell array
%           contents{get(hObject,'Value')} returns selected item from
duplafolha_vidro_exterior

```

```

% --- Executes during object creation, after setting all properties.
function duplafolha_vidro_exterior_CreateFcn(hObject, eventdata, handles)
% hObject    handle to duplafolha_vidro_exterior (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function duplafolha_rsi_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_rsi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of duplafolha_rsi as text
%         str2double(get(hObject,'String')) returns contents of duplafolha_rsi as a
double

% --- Executes during object creation, after setting all properties.
function duplafolha_rsi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to duplafolha_rsi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function duplafolha_rse_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_rse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of duplafolha_rse as text
%         str2double(get(hObject,'String')) returns contents of duplafolha_rse as a
double

% --- Executes during object creation, after setting all properties.
function duplafolha_rse_CreateFcn(hObject, eventdata, handles)
% hObject    handle to duplafolha_rse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(

```

```
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function duplafolha_rs_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_rs (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of duplafolha_rs as text
%         str2double(get(hObject,'String')) returns contents of duplafolha_rs as a
double
```

```
% --- Executes during object creation, after setting all properties.
function duplafolha_rs_CreateFcn(hObject, eventdata, handles)
% hObject      handle to duplafolha_rs (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function duplafolha_uopaco_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_uopaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of duplafolha_uopaco as text
%         str2double(get(hObject,'String')) returns contents of duplafolha_uopaco as
a double
```

```
% --- Executes during object creation, after setting all properties.
function duplafolha_uopaco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to duplafolha_uopaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function duplafolha_area_opaco_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_area_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of duplafolha_area_opaco as text
%         str2double(get(hObject,'String')) returns contents of duplafolha_area_opaco✓
as a double
```

```
% --- Executes during object creation, after setting all properties.
function duplafolha_area_opaco_CreateFcn(hObject, eventdata, handles)
% hObject    handle to duplafolha_area_opaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function duplafolha_perimetro_opaco_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_perimetro_opaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of duplafolha_perimetro_opaco as✓
text
%         str2double(get(hObject,'String')) returns contents of✓
duplafolha_perimetro_opaco as a double
```

```
% --- Executes during object creation, after setting all properties.
function duplafolha_perimetro_opaco_CreateFcn(hObject, eventdata, handles)
% hObject    handle to duplafolha_perimetro_opaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function duplafolha_linear_opaco_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_linear_opaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of duplafolha_linear_opaco as text
%         str2double(get(hObject,'String')) returns contents of✓
duplafolha_linear_opaco as a double
```

```
% --- Executes during object creation, after setting all properties.
function duplafolha_linear_opaco_CreateFcn(hObject, eventdata, handles)
% hObject    handle to duplafolha_linear_opaco (see GCBO)
```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function duplafolha_inicio_Callback(hObject, eventdata, handles)
% hObject handle to duplafolha_inicio (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('inicio.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo
Aro'))

% -----
function duplafolha_local_Callback(hObject, eventdata, handles)
% hObject handle to duplafolha_local (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('local.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo
Aro'))

% -----
function duplafolha_materiais_Callback(hObject, eventdata, handles)
% hObject handle to duplafolha_materiais (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('registamateriais.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo
Aro'))

% -----
function duplafolha_relatorios_Callback(hObject, eventdata, handles)
% hObject handle to duplafolha_relatorios (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo
Aro'))

% -----
function duplafolha_vidro_Callback(hObject, eventdata, handles)
% hObject handle to duplafolha_vidro (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('vidro.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo

```



```
Aro'))
```

```
% -----  
function duplafolha_adporta_Callback(hObject, eventdata, handles)  
% hObject    handle to duplafolha_adporta (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('Porta.m')
```

```
close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo  
Aro'))
```

```
% -----  
function duplafolha_verporta_Callback(hObject, eventdata, handles)  
% hObject    handle to duplafolha_verporta (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('porta_vereliminar.m')
```

```
close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo  
Aro'))
```

```
% -----  
function duplafolha_simples_Callback(hObject, eventdata, handles)  
% hObject    handle to duplafolha_simples (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('janela_simples.m')
```

```
close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo  
Aro'))
```

```
% -----  
function duplafolha_dupla_Callback(hObject, eventdata, handles)  
% hObject    handle to duplafolha_dupla (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
run('janela_dupla.m')
```

```
close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo  
Aro'))
```

```
% -----  
function duplafolha_verjanela_Callback(hObject, eventdata, handles)  
% hObject    handle to duplafolha_verjanela (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('janela_vereliminar.m')
```

```
close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo  
Aro'))
```

```
% -----  
function duplafolha_adhorario_Callback(hObject, eventdata, handles)  
% hObject    handle to duplafolha_adhorario (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

```

run('definir_horario.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo
Aro'))

% -----
function duplafolha_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_defhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo
Aro'))

% -----
function duplafolha_defespaco_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_defespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo
Aro'))

% -----
function duplafolha_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_espaco.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo
Aro'))

% -----
function duplafolha_adparede_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_adparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo
Aro'))

% -----
function duplafolha_verparede_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_verparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_parede.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo
Aro'))

% -----
function duplafolha_caixilho_Callback(hObject, eventdata, handles)
% hObject    handle to duplafolha_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

run('caixilho.m')

close(findobj('type','figure','name','Adicionar Janela de Dupla Folha no Mesmo Aro'))

% -----
function duplafolha_elementos_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_elementos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function duplafolha_janela_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_janela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function duplafolha_espaco_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_espaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function duplafolha_horario_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_horario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function duplafolha_porta_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_porta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function duplafolha_parede_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_parede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function duplafolha_duplafolha_Callback(hObject, eventdata, handles)
% hObject      handle to duplafolha_duplafolha (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

function varargout = local(varargin)
% LOCAL MATLAB code for local.fig
%     LOCAL, by itself, creates a new LOCAL or raises the existing
%     singleton*.
%
%     H = LOCAL returns the handle to a new LOCAL or the handle to
%     the existing singleton*.
%
%     LOCAL('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in LOCAL.M with the given input arguments.
%
%     LOCAL('Property','Value',...) creates a new LOCAL or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before local_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to local_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help local

% Last Modified by GUIDE v2.5 21-Jul-2016 04:59:49

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @local_OpeningFcn, ...
                  'gui_OutputFcn',  @local_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before local is made visible.
function local_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to local (see VARARGIN)

% Choose default command line output for local
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```

```

movegui(gcf,'center')

set(handles.figure1,'Name','Definir Local')

global carproj
%
load projecto.mat

set(handles.local_nome,'String',{'(Seleccione)'},projecto(carproj).bdados.clima.✓
nome})

% UIWAIT makes local wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = local_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in local_nome.
function local_nome_Callback(hObject, eventdata, handles)
% hObject handle to local_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns local_nome contents as✓
cell array
% contents{get(hObject,'Value')} returns selected item from local_nome

load projecto.mat
global carproj

k = get(handles.local_nome,'value');

if k==1
set(handles.local_latitude,'string','')
set(handles.local_longitude,'string','')
set(handles.local_altitude,'string','')
else
set(handles.local_latitude,'string',projecto(carproj).bdados.clima(k-1).latitude)
set(handles.local_longitude,'string',projecto(carproj).bdados.clima(k-1).longitude)
set(handles.local_altitude,'string',projecto(carproj).bdados.clima(k-1).altitude)
end

clearvars k

```

```

% --- Executes during object creation, after setting all properties.
function local_nome_CreateFcn(hObject, eventdata, handles)
% hObject    handle to local_nome (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function local_novo_latitude_Callback(hObject, eventdata, handles)
% hObject    handle to local_novo_latitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_novo_latitude as text
%         str2double(get(hObject,'String')) returns contents of local_novo_latitude
as a double

% --- Executes during object creation, after setting all properties.
function local_novo_latitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to local_novo_latitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function local_longitude_Callback(hObject, eventdata, handles)
% hObject    handle to local_longitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_longitude as text
%         str2double(get(hObject,'String')) returns contents of local_longitude as a
double

% --- Executes during object creation, after setting all properties.
function local_longitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to local_longitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
end

```

```

function local_altitude_Callback(hObject, eventdata, handles)
% hObject      handle to local_altitude (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_altitude as text
%          str2double(get(hObject,'String')) returns contents of local_altitude as a
double

```

```

% --- Executes during object creation, after setting all properties.
function local_altitude_CreateFcn(hObject, eventdata, handles)
% hObject      handle to local_altitude (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function novo_ficheiro_nome_Callback(hObject, eventdata, handles)
% hObject      handle to novo_ficheiro_nome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of novo_ficheiro_nome as text
%          str2double(get(hObject,'String')) returns contents of novo_ficheiro_nome as
a double

```

```

% --- Executes during object creation, after setting all properties.
function novo_ficheiro_nome_CreateFcn(hObject, eventdata, handles)
% hObject      handle to novo_ficheiro_nome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in local_carregar_ficheiro.
function local_carregar_ficheiro_Callback(hObject, eventdata, handles)
% hObject      handle to local_carregar_ficheiro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

addpath('base_dados_originais')
addpath('funcoes')
global carproj
load projecto.mat

choice = questdlg('Qual a extensão do ficheiro?', 'Ficheiro Climático', '.dat', '.↙
epw', '.epw');

switch choice

    case '.dat'

        filename = uigetfile(fullfile(pwd, 'base_dados_originais\dat', '*dat'));

        h=waitbar(0, 'A carregar ficheiro');
        projecto(carproj).local.nome_ficheiro=filename;
        close(h)

        h=waitbar(0.25, 'A carregar ficheiro');
        projecto(carproj).local.ficheiro = import_dat(filename);
        close(h)

        h=waitbar(0.75, 'A carregar ficheiro');
        save projecto.mat projecto
        close(h)
        h=waitbar(1, 'A carregar ficheiro');
        close(h)

    case '.epw'

        filename = uigetfile(fullfile(pwd, 'base_dados_originais\epw', '*epw'));

        h = waitbar(0, 'A carregar ficheiro');

        projecto(carproj).local.nome_ficheiro=filename;

        close(h)
        h = waitbar(0.25, 'A carregar ficheiro');
        projecto(carproj).local.ficheiro=import_epw(filename);

        close(h)
        h = waitbar(0.50, 'A carregar ficheiro');
        save projecto.mat projecto

        close(h)

end

set(handles.local_ficheiro, 'String', projecto(carproj).local.nome_ficheiro)

clearvars h choice filename

function local_tempDBV_Callback(hObject, eventdata, handles)
% hObject      handle to local_tempDBV (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```



```

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_tempDBV as text
%         str2double(get(hObject,'String')) returns contents of local_tempDBV as a double ✓

% --- Executes during object creation, after setting all properties.
function local_tempDBV_CreateFcn(hObject, eventdata, handles)
% hObject      handle to local_tempDBV (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function novo_latitude_Callback(hObject, eventdata, handles)
% hObject      handle to local_novo_latitude (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_novo_latitude as text
%         str2double(get(hObject,'String')) returns contents of local_novo_latitude ✓
as a double

% --- Executes during object creation, after setting all properties.
function novo_latitude_CreateFcn(hObject, eventdata, handles)
% hObject      handle to local_novo_latitude (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function local_novo_longitude_Callback(hObject, eventdata, handles)
% hObject      handle to local_novo_longitude (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_novo_longitude as text
%         str2double(get(hObject,'String')) returns contents of local_novo_longitude ✓
as a double

% --- Executes during object creation, after setting all properties.
function local_novo_longitude_CreateFcn(hObject, eventdata, handles)
% hObject      handle to local_novo_longitude (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function local_novo_altitude_Callback(hObject, eventdata, handles)
% hObject      handle to local_novo_altitude (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_novo_altitude as text
%         str2double(get(hObject,'String')) returns contents of local_novo_altitude
as a double

% --- Executes during object creation, after setting all properties.
function local_novo_altitude_CreateFcn(hObject, eventdata, handles)
% hObject      handle to local_novo_altitude (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function local_novo_Callback(hObject, eventdata, handles)
% hObject      handle to local_novo (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_novo as text
%         str2double(get(hObject,'String')) returns contents of local_novo as a
double

% --- Executes during object creation, after setting all properties.
function local_novo_CreateFcn(hObject, eventdata, handles)
% hObject      handle to local_novo (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
function local_tempWBV_Callback(hObject, eventdata, handles)
% hObject      handle to local_tempWBV (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_tempWBV as text
%         str2double(get(hObject,'String')) returns contents of local_tempWBV as a
double
```

```
% --- Executes during object creation, after setting all properties.
function local_tempWBV_CreateFcn(hObject, eventdata, handles)
% hObject      handle to local_tempWBV (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function local_tempDTV_Callback(hObject, eventdata, handles)
% hObject      handle to local_tempDTV (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_tempDTV as text
%         str2double(get(hObject,'String')) returns contents of local_tempDTV as a
double
```

```
% --- Executes during object creation, after setting all properties.
function local_tempDTV_CreateFcn(hObject, eventdata, handles)
% hObject      handle to local_tempDTV (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function local_tempDBI_Callback(hObject, eventdata, handles)
% hObject      handle to local_tempDBI (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_tempDBI as text
%         str2double(get(hObject,'String')) returns contents of local_tempDBI as a
double
```

```

% --- Executes during object creation, after setting all properties.
function local_tempDBI_CreateFcn(hObject, eventdata, handles)
% hObject    handle to local_tempDBI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function local_tempWBI_Callback(hObject, eventdata, handles)
% hObject    handle to local_tempWBI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_tempWBI as text
%         str2double(get(hObject,'String')) returns contents of local_tempWBI as a
double

% --- Executes during object creation, after setting all properties.
function local_tempWBI_CreateFcn(hObject, eventdata, handles)
% hObject    handle to local_tempWBI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function local_ficheiro_Callback(hObject, eventdata, handles)
% hObject    handle to local_ficheiro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of local_ficheiro as text
%         str2double(get(hObject,'String')) returns contents of local_ficheiro as a
double

% --- Executes during object creation, after setting all properties.
function local_ficheiro_CreateFcn(hObject, eventdata, handles)
% hObject    handle to local_ficheiro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in local_novo_guardar.
function local_novo_guardar_Callback(hObject, eventdata, handles)
% hObject      handle to local_novo_guardar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

load projecto.mat

global carproj

choice = questdlg('Tem a certeza que deseja adicionar o novo
local','Atenção!', 'Sim', 'Não', 'Cancelar', 'Cancelar');

switch choice
    case 'Sim'

        k = length(projecto(carproj).bdados.clima)+1;

        projecto(carproj).bdados.clima(k).nome = get(handles.local_novo, 'String');
        projecto(carproj).bdados.clima(k).latitude = get(handles.
local_novo_latitude, 'String');
        projecto(carproj).bdados.clima(k).longitude = get(handles.
local_novo_longitude, 'String');
        projecto(carproj).bdados.clima(k).altitude = get(handles.
local_novo_altitude, 'String');

        if isempty(projecto(carproj).bdados.clima(k).nome)
            projecto(carproj).bdados.clima(k).nome = 'Sem nome';
        end
        if isempty(projecto(carproj).bdados.clima(k).latitude)
            projecto(carproj).bdados.clima(k).latitude=0;
        end
        if isempty(projecto(carproj).bdados.clima(k).longitude)
            projecto(carproj).bdados.clima(k).longitude=0;
        end
        if isempty(projecto(carproj).bdados.clima(k).altitude)
            projecto(carproj).bdados.clima(k).altitude=0;
        end
        save projecto.mat projecto

        warndlg('Novo local adicionado')
        set(handles.local_nome, 'String', {'(Seleccione)', projecto(carproj).bdados.
clima.nome})

        set(handles.local_novo, 'String', '');
        set(handles.local_novo_latitude, 'String', '');
        set(handles.local_novo_longitude, 'String', '');
        set(handles.local_novo_altitude, 'String', '');

    case 'Não'
        set(handles.local_novo, 'String', '');
        set(handles.local_novo_latitude, 'String', '');

```

```

        set(handles.local_novo_longitude,'String','');
        set(handles.local_novo_altitude,'String','');
    case 'Cancelar'
end

```

```
clearvars choice k
```

```

% --- Executes on button press in adicionar_novo_local.
function adicionar_novo_local_Callback(hObject, eventdata, handles)
% hObject    handle to adicionar_novo_local (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of adicionar_novo_local

```

```

% --- Executes on button press in local_definir.
function local_definir_Callback(hObject, eventdata, handles)
% hObject    handle to local_definir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```
load projecto.mat
```

```
global carproj
```

```

choice = questdlg('Tem a certeza que deseja definir o local', 'Atenção!', 'Sim', 'Não', 'Cancelar', 'Cancelar');

```

```

switch choice
case 'Sim'
    k = get(handles.local_nome, 'Value');

    if k==1

        projecto(carproj).local.local = 'Sem local';
        projecto(carproj).local.localvalor = 0;
        projecto(carproj).local.latitude = 0;
        projecto(carproj).local.longitude = 0;
        projecto(carproj).local.altitude = 0;
    else
        projecto(carproj).local.local = projecto(carproj).bdados.clima(k-1).nome;
        projecto(carproj).local.localvalor = get(handles.local_nome, 'Value');
        projecto(carproj).local.latitude = get(handles.local_latitude, 'String');
        projecto(carproj).local.longitude = get(handles.local_longitude, 'String');
        projecto(carproj).local.altitude = get(handles.local_altitude, 'String');
    end

    projecto(carproj).local.tempDBV = get(handles.local_tempDBV, 'String');
    projecto(carproj).local.tempWBV = get(handles.local_tempWBV, 'String');
    projecto(carproj).local.tempDTV = get(handles.local_tempDTV, 'String');
    projecto(carproj).local.tempDBI = get(handles.local_tempDBI, 'String');
    projecto(carproj).local.tempWBI = get(handles.local_tempWBI, 'String');

    if isempty(projecto(carproj).local.tempDBV)
        projecto(carproj).local.tempDBV='0';
    end
end

```

```

if isempty(projecto(carproj).local.tempWBV)
    projecto(carproj).local.tempWBV='0';
end
if isempty(projecto(carproj).local.tempDTV)
    projecto(carproj).local.tempDTV='0';
end
if isempty(projecto(carproj).local.tempDBI)
    projecto(carproj).local.tempDBI='0';
end
if isempty(projecto(carproj).local.tempWBI)
    projecto(carproj).local.tempWBI='0';
end

if isfield(projecto(carproj).local,'ficheiro')==0
    projecto(carproj).local.ficheiro = zeros(8760,8);
end

save projecto.mat projecto

warndlg('Local definido')

case 'Não'
    if isempty(projecto(carproj).local)
        set(handles.local_nome,'Value',1)
        set(handles.local_latitude,'String','');
        set(handles.local_longitude,'String','');
        set(handles.local_altitude,'String','');
        set(handles.local_tempDBV,'String','');
        set(handles.local_tempWBV,'String','');
        set(handles.local_tempDTV,'String','');
        set(handles.local_tempDBI,'String','');
        set(handles.local_tempWBI,'String','');
        set(handles.local_ficheiro,'String','');
    else
        set(handles.local_nome,'Value',projecto(carproj).local.localvalor);
        set(handles.local_latitude,'String',projecto(carproj).local.latitude);
        set(handles.local_longitude,'String',projecto(carproj).local.longitude);
        set(handles.local_altitude,'String',projecto(carproj).local.altitude);
        set(handles.local_tempDBV,'String',projecto(carproj).local.tempDBV);
        set(handles.local_tempWBV,'String',projecto(carproj).local.tempWBV);
        set(handles.local_tempDTV,'String',projecto(carproj).local.tempDTV);
        set(handles.local_tempDBI,'String',projecto(carproj).local.tempDBI);
        set(handles.local_tempWBI,'String',projecto(carproj).local.tempWBI);
        set(handles.local_ficheiro,'String',projecto(carproj).local.nome_ficheiro);
    end
case 'Cancelar'
end

clearvars choice k

% --- Executes on button press in local_eliminar.
function local_eliminar_Callback(hObject, eventdata, handles)
% hObject    handle to local_eliminar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

load projecto.mat

global carproj

choice = questdlg('Tem a certeza que deseja eliminar o ✓
projecto', 'Atenção!', 'Sim', 'Não', 'Cancelar', 'Cancelar');

    switch choice
        case 'Sim'
            k = get(handles.local_nome, 'Value');

            if k>21

                projecto(carproj).bdados.clima(k-1)=[];

            save projecto.mat projecto

            set(handles.local_nome, 'String', {'(Seleccione)', projecto ✓
(carproj).bdados.clima.nome})

            warndlg('Local eliminado')

            if isempty(projecto(carproj).local)
                set(handles.local_nome, 'Value', 1)
                set(handles.local_latitude, 'String', '');
                set(handles.local_longitude, 'String', '');
                set(handles.local_altitude, 'String', '');
                set(handles.local_tempDBV, 'String', '');
                set(handles.local_tempWBV, 'String', '');
                set(handles.local_tempDTV, 'String', '');
                set(handles.local_tempDBI, 'String', '');
                set(handles.local_tempWBI, 'String', '');
                set(handles.local_ficheiro, 'String', '');
            else
                set(handles.local_nome, 'Value', projecto(carproj).local. ✓
localvalor);
                set(handles.local_latitude, 'String', projecto(carproj).local. ✓
latitude);
                set(handles.local_longitude, 'String', projecto(carproj). ✓
local.longitude);
                set(handles.local_altitude, 'String', projecto(carproj).local. ✓
altitude);
                set(handles.local_tempDBV, 'String', projecto(carproj).local. ✓
tempDBV);
                set(handles.local_tempWBV, 'String', projecto(carproj).local. ✓
tempWBV);
                set(handles.local_tempDTV, 'String', projecto(carproj).local. ✓
tempDTV);
                set(handles.local_tempDBI, 'String', projecto(carproj).local. ✓
tempDBI);
                set(handles.local_tempWBI, 'String', projecto(carproj).local. ✓
tempWBI);
                set(handles.local_ficheiro, 'String', projecto(carproj).local. ✓
nome_ficheiro);
            end
        else
            errordlg('Não é possível eliminar este local', 'Erro', 'modal')
        end
end

```



```

        case 'Não'

        case 'Cancelar'
    end

    clearvars choice k

% --- Executes when figure1 is resized.
function figure1_ResizeFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function local_inicio_Callback(hObject, eventdata, handles)
% hObject    handle to local_inicio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('inicio.m')

close(findobj('type','figure','name','Definir Local'))

% -----
function local_local_Callback(hObject, eventdata, handles)
% hObject    handle to local_local (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function local_materiais_Callback(hObject, eventdata, handles)
% hObject    handle to local_materiais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('registamateriais.m')

close(findobj('type','figure','name','Definir Local'))

% -----
function local_elementos_Callback(hObject, eventdata, handles)
% hObject    handle to local_elementos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function local_espaco_Callback(hObject, eventdata, handles)
% hObject    handle to local_espaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% -----
function local_relatorios_Callback(hObject, eventdata, handles)
% hObject      handle to local_relatorios (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('relatorios.m')
close(findobj('type','figure','name','Definir Local'))

% --- Executes during object creation, after setting all properties.
function local_latitude_CreateFcn(hObject, eventdata, handles)
% hObject      handle to local_latitude (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function local_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to local_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('vidro.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_caixilho_Callback(hObject, eventdata, handles)
% hObject      handle to local_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('caixilho.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_adiporta_Callback(hObject, eventdata, handles)
% hObject      handle to local_adiporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('Porta.m')
close(findobj('type','figure','name','Definir Local'))

```

```

% -----
function local_verporta_Callback(hObject, eventdata, handles)
% hObject    handle to local_verporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('porta_vereliminair.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_jansimples_Callback(hObject, eventdata, handles)
% hObject    handle to local_jansimples (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('janela_simples.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_jandupla_Callback(hObject, eventdata, handles)
% hObject    handle to local_jandupla (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('janela_dupla.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_duplafolha_Callback(hObject, eventdata, handles)
% hObject    handle to local_duplafolha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('janela_duplafolhamesmoaro.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_verjanela_Callback(hObject, eventdata, handles)
% hObject    handle to local_verjanela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('janela_vereliminair.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_horario_Callback(hObject, eventdata, handles)
% hObject    handle to local_horario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('definir_horario.m')
close(findobj('type','figure','name','Definir Local'))

```

```

% -----
function local_defhorario_Callback(hObject, eventdata, handles)
% hObject      handle to local_defhorario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('horario.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_defespaco_Callback(hObject, eventdata, handles)
% hObject      handle to local_defespaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('definir_espaco.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_verespaco_Callback(hObject, eventdata, handles)
% hObject      handle to local_verespaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('ver_espaco.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_adparede_Callback(hObject, eventdata, handles)
% hObject      handle to local_adparede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('adicionarparede.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_verparede_Callback(hObject, eventdata, handles)
% hObject      handle to local_verparede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('ver_parede.m')
close(findobj('type','figure','name','Definir Local'))

% -----
function local_hor_Callback(hObject, eventdata, handles)
% hObject      handle to local_hor (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function local_janela_Callback(hObject, eventdata, handles)
% hObject      handle to local_janela (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function local_porta_Callback(hObject, eventdata, handles)
% hObject      handle to local_porta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function local_paredes_Callback(hObject, eventdata, handles)
% hObject      handle to local_paredes (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```

function varargout = Porta(varargin)
% PORTA MATLAB code for Porta.fig
%     PORTA, by itself, creates a new PORTA or raises the existing
%     singleton*.
%
%     H = PORTA returns the handle to a new PORTA or the handle to
%     the existing singleton*.
%
%     PORTA('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in PORTA.M with the given input arguments.
%
%     PORTA('Property','Value',...) creates a new PORTA or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before Porta_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to Porta_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Porta

% Last Modified by GUIDE v2.5 21-Jul-2016 06:13:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Porta_OpeningFcn, ...
                  'gui_OutputFcn',  @Porta_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Porta is made visible.
function Porta_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Porta (see VARARGIN)

% Choose default command line output for Porta
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```

```

movegui(gcf,'center')

set(handles.figure1,'Name','Adicionar Porta')

load projecto.mat

global carproj

set(handles.porta_popvidro,'String',{'(Selecione)',projecto(carproj).bdados.vidro.✓
nome})
set(handles.porta_popcaixilho,'String',{'(Selecione)',projecto(carproj).bdados.✓
caixilho.nome})

% UIWAIT makes Porta wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Porta_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function porta_nome_Callback(hObject, eventdata, handles)
% hObject handle to porta_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of porta_nome as text
% str2double(get(hObject,'String')) returns contents of porta_nome as a ✓
double

% --- Executes during object creation, after setting all properties.
function porta_nome_CreateFcn(hObject, eventdata, handles)
% hObject handle to porta_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in porta_imagem_areas.
function porta_imagem_areas_Callback(hObject, eventdata, handles)
% hObject handle to porta_imagem_areas (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in porta_imagem_portas.
function porta_imagem_portas_Callback(hObject, eventdata, handles)
% hObject      handle to porta_imagem_portas (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in porta_guardar.
function porta_guardar_Callback(hObject, eventdata, handles)
% hObject      handle to porta_guardar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
load projecto.mat

global carproj

choice = questdlg('Deseja adicionar a porta?', 'Adicionar✓
Porta', 'Sim', 'Não', 'Cancelar', 'Cancelar');

switch choice

    case 'Sim'

        k = length(projecto(carproj).bdados.porta)+1;

        if get(handles.porta_check_uporta, 'Value')==1

            projecto(carproj).bdados.porta(k).nome = get(handles.porta_nome, 'String');
            projecto(carproj).bdados.porta(k).ud = str2double(get(handles.✓
porta_uporta, 'String'));

            save projecto.mat projecto
            warndlg('Porta adicionada', 'Aviso !')
        else
            projecto(carproj).bdados.porta(k).nome = get(handles.porta_nome, 'String');
            y1 = get(handles.porta_popvidro, 'Value');
            x1 = str2double(projecto(carproj).bdados.vidro(y1-1).ug);
            x2 = str2double(get(handles.porta_area_vidro, 'String'));
            y3 = get(handles.porta_popcaixilho, 'Value');
            x3 = str2double(projecto(carproj).bdados.caixilho(y3-1).uf);
            x4 = str2double(get(handles.porta_area_caixilho, 'String'));
            x5 = str2double(get(handles.porta_perimetro_vidro, 'String'));
            x6 = str2double(get(handles.porta_linear_vidro, 'String'));

            if get(handles.porta_check_opaco, 'Value')==1
                x7 = str2double(get(handles.porta_coef_opaco, 'String'));
                x8 = str2double(get(handles.porta_area_opaco, 'String'));
                x9 = str2double(get(handles.porta_perimetro_opaco, 'String'));
                x10 = str2double(get(handles.porta_linear_opaco, 'String'));
            else
                x7=0;
                x8=0;
                x9=0;
                x10=0;
            end

            ud = ((x2*x1)+(x8*x7)+(x4*x3)+(x5*x6)+(x9*x10))/(x2+x4+x8);

```



```

projecto(carproj).bdados.porta(k).nome = get(handles.porta_nome, 'String');
projecto(carproj).bdados.porta(k).ud = ud;

save projecto.mat projecto
warndlg('Porta adicionada', 'Aviso !')
end

case 'Não'
case 'Cancelar'
end

clearvars choice k y1 y3 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 ud

% --- Executes on button press in porta_check_uporta.
function porta_check_uporta_Callback(hObject, eventdata, handles)
% hObject    handle to porta_check_uporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject, 'Value') returns toggle state of porta_check_uporta
if get(handles.porta_check_uporta, 'Value')==1
    set(handles.porta_upanel, 'Visible', 'On')
else
    set(handles.porta_upanel, 'Visible', 'Off')
end

% --- Executes on button press in porta_check_opaco.
function porta_check_opaco_Callback(hObject, eventdata, handles)
% hObject    handle to porta_check_opaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject, 'Value') returns toggle state of porta_check_opaco

if get(handles.porta_check_opaco, 'Value')==1
    set(handles.porta_panel_opaco, 'Visible', 'On')
else
    set(handles.porta_panel_opaco, 'Visible', 'Off')
end

function porta_uporta_Callback(hObject, eventdata, handles)
% hObject    handle to porta_uporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of porta_uporta as text
%        str2double(get(hObject, 'String')) returns contents of porta_uporta as a
double

% --- Executes during object creation, after setting all properties.
function porta_uporta_CreateFcn(hObject, eventdata, handles)
% hObject    handle to porta_uporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(

```

```

(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in porta_popvidro.
function porta_popvidro_Callback(hObject, eventdata, handles)
% hObject      handle to porta_popvidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns porta_popvidro contents ✓
as cell array
%           contents{get(hObject,'Value')} returns selected item from porta_popvidro

% --- Executes during object creation, after setting all properties.
function porta_popvidro_CreateFcn(hObject, eventdata, handles)
% hObject      handle to porta_popvidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function porta_area_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to porta_area_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of porta_area_vidro as text
%         str2double(get(hObject,'String')) returns contents of porta_area_vidro as a ✓
double

% --- Executes during object creation, after setting all properties.
function porta_area_vidro_CreateFcn(hObject, eventdata, handles)
% hObject      handle to porta_area_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in porta_popcaixilho.
function porta_popcaixilho_Callback(hObject, eventdata, handles)
% hObject      handle to porta_popcaixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject,'String')) returns porta_popcaixilho✓
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from porta_popcaixilho

% --- Executes during object creation, after setting all properties.
function porta_popcaixilho_CreateFcn(hObject, eventdata, handles)
% hObject    handle to porta_popcaixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function porta_area_caixilho_Callback(hObject, eventdata, handles)
% hObject    handle to porta_area_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of porta_area_caixilho as text
%         str2double(get(hObject,'String')) returns contents of porta_area_caixilho✓
as a double

% --- Executes during object creation, after setting all properties.
function porta_area_caixilho_CreateFcn(hObject, eventdata, handles)
% hObject    handle to porta_area_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function porta_perimetro_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to porta_perimetro_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of porta_perimetro_vidro as text
%         str2double(get(hObject,'String')) returns contents of porta_perimetro_vidro✓
as a double

% --- Executes during object creation, after setting all properties.
function porta_perimetro_vidro_CreateFcn(hObject, eventdata, handles)
% hObject    handle to porta_perimetro_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function porta_linear_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to porta_linear_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of porta_linear_vidro as text
%      str2double(get(hObject,'String')) returns contents of porta_linear_vidro as a
a double

% --- Executes during object creation, after setting all properties.
function porta_linear_vidro_CreateFcn(hObject, eventdata, handles)
% hObject      handle to porta_linear_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function porta_coef_opaco_Callback(hObject, eventdata, handles)
% hObject      handle to porta_coef_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of porta_coef_opaco as text
%      str2double(get(hObject,'String')) returns contents of porta_coef_opaco as a
double

% --- Executes during object creation, after setting all properties.
function porta_coef_opaco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to porta_coef_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function porta_area_opaco_Callback(hObject, eventdata, handles)
% hObject      handle to porta_area_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of porta_area_opaco as text
%         str2double(get(hObject,'String')) returns contents of porta_area_opaco as a
double

% --- Executes during object creation, after setting all properties.
function porta_area_opaco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to porta_area_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function porta_perimetro_opaco_Callback(hObject, eventdata, handles)
% hObject      handle to porta_perimetro_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of porta_perimetro_opaco as text
%         str2double(get(hObject,'String')) returns contents of porta_perimetro_opaco
as a double

% --- Executes during object creation, after setting all properties.
function porta_perimetro_opaco_CreateFcn(hObject, eventdata, handles)
% hObject      handle to porta_perimetro_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function porta_linear_opaco_Callback(hObject, eventdata, handles)
% hObject      handle to porta_linear_opaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of porta_linear_opaco as text
%         str2double(get(hObject,'String')) returns contents of porta_linear_opaco as
a double

```

```

% --- Executes during object creation, after setting all properties.
function porta_linear_opaco_CreateFcn(hObject, eventdata, handles)
% hObject    handle to porta_linear_opaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function porta_local_Callback(hObject, eventdata, handles)
% hObject    handle to porta_local (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('local.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_materiais_Callback(hObject, eventdata, handles)
% hObject    handle to porta_materiais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('registamateriais.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_relatorios_Callback(hObject, eventdata, handles)
% hObject    handle to porta_relatorios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_Callback(hObject, eventdata, handles)
% hObject    handle to porta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function porta_verporta_Callback(hObject, eventdata, handles)
% hObject    handle to porta_verporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('porta_vereliminar.m')

close(findobj('type','figure','name','Adicionar Porta'))

```

```

% -----
function porta_simples_Callback(hObject, eventdata, handles)
% hObject      handle to porta_simples (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_simples.m')

close(findobj('type','figure','name','Adicionar Porta'))
% -----
function porta_dupla_Callback(hObject, eventdata, handles)
% hObject      handle to porta_dupla (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_dupla.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_duplafolha_Callback(hObject, eventdata, handles)
% hObject      handle to porta_duplafolha (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_verjanela_Callback(hObject, eventdata, handles)
% hObject      handle to porta_verjanela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_adhorario_Callback(hObject, eventdata, handles)
% hObject      handle to porta_adhorario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('definir_horario.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_defhorario_Callback(hObject, eventdata, handles)
% hObject      handle to porta_defhorario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_defespaco_Callback(hObject, eventdata, handles)
% hObject      handle to porta_defespaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

```

```

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to porta_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_espaco.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_adparede_Callback(hObject, eventdata, handles)
% hObject    handle to porta_adparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_verparede_Callback(hObject, eventdata, handles)
% hObject    handle to porta_verparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_parede.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_inicio_Callback(hObject, eventdata, handles)
% hObject    handle to porta_inicio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('inicio.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to porta_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('vidro.m')

close(findobj('type','figure','name','Adicionar Porta'))

% -----
function porta_caixilho_Callback(hObject, eventdata, handles)
% hObject    handle to porta_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('caixilho.m')

close(findobj('type','figure','name','Adicionar Porta'))

```



```

% -----
function porta_elementos_Callback(hObject, eventdata, handles)
% hObject    handle to porta_elementos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function porta_espaco_Callback(hObject, eventdata, handles)
% hObject    handle to porta_espaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function porta_horario_Callback(hObject, eventdata, handles)
% hObject    handle to porta_horario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function porta_janela_Callback(hObject, eventdata, handles)
% hObject    handle to porta_janela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function porta_porta_Callback(hObject, eventdata, handles)
% hObject    handle to porta_porta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function porta_parede_Callback(hObject, eventdata, handles)
% hObject    handle to porta_parede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

function varargout = porta_vereliminar(varargin)
% PORTA_VERELIMINAR MATLAB code for porta_vereliminar.fig
%     PORTA_VERELIMINAR, by itself, creates a new PORTA_VERELIMINAR or raises the
existing
%     singleton*.
%
%     H = PORTA_VERELIMINAR returns the handle to a new PORTA_VERELIMINAR or the
handle to
%     the existing singleton*.
%
%     PORTA_VERELIMINAR('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in PORTA_VERELIMINAR.M with the given input
arguments.
%
%     PORTA_VERELIMINAR('Property','Value',...) creates a new PORTA_VERELIMINAR or
raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before porta_vereliminar_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to porta_vereliminar_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help porta_vereliminar

% Last Modified by GUIDE v2.5 21-Jul-2016 11:42:47

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @porta_vereliminar_OpeningFcn, ...
                  'gui_OutputFcn',  @porta_vereliminar_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before porta_vereliminar is made visible.
function porta_vereliminar_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to porta_vereliminar (see VARARGIN)

% Choose default command line output for porta_vereliminar
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

movegui(gcf, 'center')

set(handles.figure1, 'Name', 'Ver / Eliminar Porta')

load projecto.mat

global carproj

set(handles.verporta_lista, 'String', {projecto(carproj).bdados.porta.nome})

% UIWAIT makes porta_vereliminar wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = porta_vereliminar_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in verporta_eliminar.
function verporta_eliminar_Callback(hObject, eventdata, handles)
% hObject handle to verporta_eliminar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

if get(handles.verporta_lista, 'Value') > 5

choice = questdlg('Deseja eliminar a porta?', 'Eliminar Porta', 'Sim', 'Não', 'Cancelar', 'Cancelar');

switch choice
case 'Sim'
load projecto.mat

global carproj

k = get(handles.verporta_lista, 'Value');

projecto(carproj).bdados.porta(k) = [];

save projecto.mat projecto

set(handles.verporta_lista, 'Value', 1)
load projecto.mat
set(handles.verporta_lista, 'String', {projecto(carproj).bdados.
porta.nome})

set(handles.verporta_nome, 'String', '')

```

```

        set(handles.verporta_u, 'String', '')

        warndlg('Porta eliminada', 'Aviso !')

        case 'Não'
        case 'Cancelar'
    end

else

    warndlg('Não é possível eliminar a porta', 'Aviso !')
end

clearvars choice k

% --- Executes on button press in verporta_ver.
function verporta_ver_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_ver (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

load projecto.mat

global carproj

k = get(handles.verporta_lista, 'Value');

set(handles.verporta_nome, 'String', projecto(carproj).bdados.porta(k).nome)
set(handles.verporta_u, 'String', projecto(carproj).bdados.porta(k).ud)

clearvars k

% --- Executes on selection change in verporta_lista.
function verporta_lista_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_lista (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String')) returns verporta_lista contents ✓
% as cell array
%         contents{get(hObject, 'Value')} returns selected item from verporta_lista

% --- Executes during object creation, after setting all properties.
function verporta_lista_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verporta_lista (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function verporta_nome_Callback(hObject, eventdata, handles)

```

```
% hObject    handle to verporta_nome (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verporta_nome as text
%        str2double(get(hObject,'String')) returns contents of verporta_nome as a
double
```

```
% --- Executes during object creation, after setting all properties.
function verporta_nome_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verporta_nome (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function verporta_tipo_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of verporta_tipo as text
%        str2double(get(hObject,'String')) returns contents of verporta_tipo as a
double
```

```
% --- Executes during object creation, after setting all properties.
function verporta_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verporta_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function verporta_u_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_u (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of verporta_u as text
%        str2double(get(hObject,'String')) returns contents of verporta_u as a
double
```

```
% --- Executes during object creation, after setting all properties.
```

```

function verporta_u_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verporta_u (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function verporta_inicio_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_inicio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('inicio.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----
function verporta_local_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_local (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('local.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----
function verporta_materiais_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_materiais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('registamateriais.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----
function verporta_relatorios_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_relatorios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----
function verporta_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('vidro.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----

```

```
function verporta_caixilho_Callback(hObject, eventdata, handles)
% hObject      handle to verporta_caixilho (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('caixilho.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Porta'))
```

```
% -----
function verporta_adporta_Callback(hObject, eventdata, handles)
% hObject      handle to verporta_adporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('Porta.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Porta'))
```

```
% -----
function verporta_Callback(hObject, eventdata, handles)
% hObject      handle to verporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function verporta_simples_Callback(hObject, eventdata, handles)
% hObject      handle to verporta_simples (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_simples.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Porta'))
```

```
% -----
function verporta_dupla_Callback(hObject, eventdata, handles)
% hObject      handle to verporta_dupla (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_dupla.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Porta'))
```

```
% -----
function verporta_duplafolha_Callback(hObject, eventdata, handles)
% hObject      handle to verporta_duplafolha (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Porta'))
```

```
% -----
function verjanela_verporta_Callback(hObject, eventdata, handles)
% hObject      handle to verjanela_verporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Porta'))
```

```

% -----
function verporta_adhorario_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_adhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_horario.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----
function verporta_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_defhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----
function verporta_defespaco_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_defespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----
function verporta_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_espaco.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----
function verporta_adparede_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_adparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----
function verporta_verparede_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_verparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_parede.m')

close(findobj('type','figure','name','Ver / Eliminar Porta'))

% -----
function verporta_elementos_Callback(hObject, eventdata, handles)
% hObject    handle to verporta_elementos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```



```
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
```

```
function verporta_espaco_Callback(hObject, eventdata, handles)
% hObject      handle to verporta_espaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
```

```
function verporta_horario_Callback(hObject, eventdata, handles)
% hObject      handle to verporta_horario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
```

```
function verporta_janela_Callback(hObject, eventdata, handles)
% hObject      handle to verporta_janela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
```

```
function verporta_porta_Callback(hObject, eventdata, handles)
% hObject      handle to verporta_porta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
```

```
function verporta_parede_Callback(hObject, eventdata, handles)
% hObject      handle to verporta_parede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```

function varargout = registamateriais(varargin)
% REGISTAMATERIAIS MATLAB code for registamateriais.fig
%     REGISTAMATERIAIS, by itself, creates a new REGISTAMATERIAIS or raises the
existing
%     singleton*.
%
%     H = REGISTAMATERIAIS returns the handle to a new REGISTAMATERIAIS or the
handle to
%     the existing singleton*.
%
%     REGISTAMATERIAIS('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in REGISTAMATERIAIS.M with the given input arguments.
%
%     REGISTAMATERIAIS('Property','Value',...) creates a new REGISTAMATERIAIS or
raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before registamateriais_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to registamateriais_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help registamateriais

% Last Modified by GUIDE v2.5 21-Jul-2016 05:02:37

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @registamateriais_OpeningFcn, ...
                  'gui_OutputFcn',  @registamateriais_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before registamateriais is made visible.
function registamateriais_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to registamateriais (see VARARGIN)

% Choose default command line output for registamateriais
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

movegui(gcf, 'center')

set(handles.figure1, 'Name', 'Materiais')

load projecto.mat
global carproj

    if carproj >0

w = {projecto(carproj).bdados.material.agua(1:end).nome,projecto(carproj).bdados.✓
material.argamassa(1:end).nome,projecto(carproj).bdados.material.betao(1:end).✓
nome,...
    projecto(carproj).bdados.material.borrachas(1:end).nome,projecto(carproj).✓
bdados.material.fibrocimento(1:end).nome,...
    projecto(carproj).bdados.material.gases(1:end).nome,projecto(carproj).bdados.✓
material.gessos(1:end).nome,projecto(carproj).bdados.material.✓
impermeabilizacao_mastiques(1:end).nome,...
    projecto(carproj).bdados.material.inertes_solos_terras(end).nome,projecto✓
(carproj).bdados.material.isolamento(1:end).nome,...
    projecto(carproj).bdados.material.madeira(1:end).nome,projecto(carproj).bdados.✓
material.plasticos(1:end).nome,projecto(carproj).bdados.material.ceramico(1:end).✓
nome,...
    projecto(carproj).bdados.material.metais(1:end).nome,projecto(carproj).bdados.✓
material.pedras(1:end).nome,projecto(carproj).bdados.material.✓
revestimentos_pisos_paredes(1:end).nome,...
    projecto(carproj).bdados.material.vidros(1:end).nome};

set(handles.materiais_lista, 'String', w)

    else

    end

clearvars w

% UIWAIT makes registamateriais wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = registamateriais_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in materiais_tipo.

```

```

function materiais_tipo_Callback(hObject, eventdata, handles)
% hObject      handle to materiais_tipo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns materiais_tipo contents ✓
as cell array
%           contents{get(hObject,'Value')} returns selected item from materiais_tipo ✓

load projecto.mat
global carproj

switch get(handles.materiais_tipo,'value')
    case 1

        w = {projecto(carproj).bdados.material.agua(1:end).nome,projecto(carproj). ✓
bdados.material.argamassa(1:end).nome,projecto(carproj).bdados.material.betao(1: ✓
end).nome,...
        projeto(carproj).bdados.material.borrachas(1:end).nome,projecto(carproj). ✓
bdados.material.fibrocimento(1:end).nome,...
        projeto(carproj).bdados.material.gases(1:end).nome,projecto(carproj).bdados. ✓
material.gessos(1:end).nome,projecto(carproj).bdados.material. ✓
impermeabilizacao_mastiques(1:end).nome,...
        projeto(carproj).bdados.material.inertes_solos Terras(end).nome,projecto ✓
(carproj).bdados.material.isolamento(1:end).nome,...
        projeto(carproj).bdados.material.madeira(1:end).nome,projecto(carproj).bdados. ✓
material.plasticos(1:end).nome,projecto(carproj).bdados.material.ceramico(1:end). ✓
nome,...
        projeto(carproj).bdados.material.metais(1:end).nome,projecto(carproj).bdados. ✓
material.pedras(1:end).nome,projecto(carproj).bdados.material. ✓
revestimentos_pisos_paredes(1:end).nome,...
        projeto(carproj).bdados.material.vidros(1:end).nome};

        set(handles.materiais_lista,'String',w)

set(handles.materiais_nome,'string','')
set(handles.materiais_rho,'string','')
set(handles.materiais_k,'string','')

    case 2
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.agua ✓
(1:end).nome})
    case 3
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material. ✓
argamassa(1:end).nome})
    case 4
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material. ✓
betao(1:end).nome})
    case 5
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material. ✓
borrachas(1:end).nome})
    case 6
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material. ✓
fibrocimento(1:end).nome})
    case 7
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material. ✓
gases(1:end).nome})

```

```

    case 8
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
gessos(1:end).nome})
    case 9
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
impermeabilizacao_mastiques(1:end).nome})
    case 10
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
inertes_solos_terras(1:end).nome})
    case 11
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
isolamento(1:end).nome})
    case 12
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
madeira(1:end).nome})
    case 13
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
plasticos(1:end).nome})
    case 14
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
ceramico(1:end).nome})
    case 15
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
metais(1:end).nome})
    case 16
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
pedras(1:end).nome})
    case 17
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
revestimentos_pisos_paredes(1:end).nome})
    case 18
        set(handles.materiais_lista,'string',{projecto(carproj).bdados.material.✓
vidros(1:end).nome})
end

```

```

set(handles.materiais_lista,'value',1)

```

```

clearvars w

```

```

% --- Executes during object creation, after setting all properties.
function materiais_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to materiais_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function materiais_nome_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_nome (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of materiais_nome as text
%         str2double(get(hObject,'String')) returns contents of materiais_nome as a
double

% --- Executes during object creation, after setting all properties.
function materiais_nome_CreateFcn(hObject, eventdata, handles)
% hObject    handle to materiais_nome (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function materiais_rho_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_rho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of materiais_rho as text
%         str2double(get(hObject,'String')) returns contents of materiais_rho as a
double

% --- Executes during object creation, after setting all properties.
function materiais_rho_CreateFcn(hObject, eventdata, handles)
% hObject    handle to materiais_rho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function materiais_k_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_k (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of materiais_k as text
%         str2double(get(hObject,'String')) returns contents of materiais_k as a
double

% --- Executes during object creation, after setting all properties.
function materiais_k_CreateFcn(hObject, eventdata, handles)
% hObject    handle to materiais_k (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in materiais_registar.
function materiais_registar_Callback(hObject, eventdata, handles)
% hObject      handle to materiais_registar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

load projecto.mat

global carproj

switch get(handles.materiais_tipo,'value');
    case 1 %(seleccione)

        case 2 %Água

            k = length(projecto(carproj).bdados.material.agua)+1;
            projecto(carproj).bdados.material.agua(k).nome = get(handles.
materiais_nome,'string');
            projecto(carproj).bdados.material.agua(k).classificacao = 'Água';
            projecto(carproj).bdados.material.agua(k).rho = str2double(get(handles.
materiais_rho,'string'));
            projecto(carproj).bdados.material.agua(k).k = str2double(get(handles.
materiais_k,'string'));

            if isempty(projecto(carproj).bdados.material.agua(k).nome)
                projecto(carproj).bdados.material.agua(k).nome = 'Sem nome';
            end
            if isnan(projecto(carproj).bdados.material.agua(k).rho)
                projecto(carproj).bdados.material.agua(k).rho = '0';
            end
            if isnan( projecto(carproj).bdados.material.agua(k).k)
                projecto(carproj).bdados.material.agua(k).k = '0';
            end

        case 3 %Argamassa

            k = length(projecto(carproj).bdados.material.argamassa) + 1;
            projecto(carproj).bdados.material.argamassa(k).nome = get(handles.
materiais_nome,'string');
            projecto(carproj).bdados.material.argamassa(k).classificacao = 'Argamassa';
            projecto(carproj).bdados.material.argamassa(k).rho = str2double( get
(handles.materiais_rho,'string') );
            projecto(carproj).bdados.material.argamassa(k).k = str2double( get(handles.
materiais_k,'string') );

            if isempty(projecto(carproj).bdados.material.argamassa(k).nome)
                projecto(carproj).bdados.material.argamassa(k).nome = 'Sem nome';
            end
            if isnan(projecto(carproj).bdados.material.argamassa(k).rho)
                projecto(carproj).bdados.material.argamassa(k).rho = '0';
            end
end

```

```

        if isnan( projecto(carproj).bdados.material.argamassa(k).k)
            projecto(carproj).bdados.material.argamassa(k).k = '0';
        end

case 4 %Betão

    k = length(projecto(carproj).bdados.material.betao) + 1;
    projecto(carproj).bdados.material.betao(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.betao(k).classificacao = 'Betão';
    projecto(carproj).bdados.material.betao(k).rho = str2double( get(handles.✓
materiais_rho, 'string') );
    projecto(carproj).bdados.material.betao(k).k = str2double( get(handles.✓
materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.betao(k).nome)
        projecto(carproj).bdados.material.betao(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.betao(k).rho)
        projecto(carproj).bdados.material.betao(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.betao(k).k)
        projecto(carproj).bdados.material.betao(k).k = '0';
    end

case 5 %Borrachas

    k = length(projecto(carproj).bdados.material.borrachas) + 1;
    projecto(carproj).bdados.material.borrachas(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.borrachas(k).classificacao = 'Borrachas';
    projecto(carproj).bdados.material.borrachas(k).rho = str2double( get✓
(handles.materiais_rho, 'string') );
    projecto(carproj).bdados.material.borrachas(k).k = str2double( get(handles.✓
materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.borrachas(k).nome)
        projecto(carproj).bdados.material.borrachas(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.borrachas(k).rho)
        projecto(carproj).bdados.material.borrachas(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.borrachas(k).k)
        projecto(carproj).bdados.material.borrachas(k).k = '0';
    end

case 6 %Fibrocimento

    k = length(projecto(carproj).bdados.material.fibrocimento) + 1;
    projecto(carproj).bdados.material.fibrocimento(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.fibrocimento(k).classificacao =✓
'Fibrocimento';
    projecto(carproj).bdados.material.fibrocimento(k).rho = str2double( get✓
(handles.materiais_rho, 'string') );
    projecto(carproj).bdados.material.fibrocimento(k).k = str2double( get✓
(handles.materiais_k, 'string') );

```



```

if isempty(projecto(carproj).bdados.material.fibrocimento(k).nome)
    projecto(carproj).bdados.material.fibrocimento(k).nome = 'Sem nome';
end
if isnan(projecto(carproj).bdados.material.fibrocimento(k).rho)
    projecto(carproj).bdados.material.fibrocimento(k).rho = '0';
end
if isnan( projecto(carproj).bdados.material.fibrocimento(k).k)
    projecto(carproj).bdados.material.fibrocimento(k).k = '0';
end

case 7 %Gases

    k = length(projecto(carproj).bdados.material.gases) + 1;
    projecto(carproj).bdados.material.gases(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.gases(k).classificacao = 'Gases';
    projecto(carproj).bdados.material.gases(k).rho = str2double( get(handles.✓
materiais_rho, 'string') );
    projecto(carproj).bdados.material.gases(k).k = str2double( get(handles.✓
materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.gases(k).nome)
        projecto(carproj).bdados.material.gases(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.gases(k).rho)
        projecto(carproj).bdados.material.gases(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.gases(k).k)
        projecto(carproj).bdados.material.gases(k).k = '0';
    end

case 8 %Gessos

    k = length(projecto(carproj).bdados.material.gessos) + 1;
    projecto(carproj).bdados.material.gessos(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.gessos(k).classificacao = 'Gessos';
    projecto(carproj).bdados.material.gessos(k).rho = str2double( get(handles.✓
materiais_rho, 'string') );
    projecto(carproj).bdados.material.gessos(k).k = str2double( get(handles.✓
materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.gessos(k).nome)
        projecto(carproj).bdados.material.gessos(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.gessos(k).rho)
        projecto(carproj).bdados.material.gessos(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.gessos(k).k)
        projecto(carproj).bdados.material.gessos(k).k = '0';
    end

case 9 %Impermeabilização e Mastiques

    k = length(projecto(carproj).bdados.material.impermeabilizacao_mastiques) +✓
1;
    projecto(carproj).bdados.material.impermeabilizacao_mastiques(k).nome = get✓

```

```

(handles.materiais_nome, 'string');
    projecto(carproj).bdados.material.impermeabilizacao_mastiques(k).✓
classificacao = 'Materiais de impermeabilização e mastiques';
    projecto(carproj).bdados.material.impermeabilizacao_mastiques(k).rho =✓
str2double( get(handles.materiais_rho, 'string') );
    projecto(carproj).bdados.material.impermeabilizacao_mastiques(k).k =✓
str2double( get(handles.materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.impermeabilizacao_mastiques(k).✓
nome)
        projecto(carproj).bdados.material.impermeabilizacao_mastiques(k).nome =✓
'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.impermeabilizacao_mastiques(k).✓
rho)
        projecto(carproj).bdados.material.impermeabilizacao_mastiques(k).rho =✓
'0';
    end
    if isnan( projecto(carproj).bdados.material.impermeabilizacao_mastiques(k).✓
k)
        projecto(carproj).bdados.material.impermeabilizacao_mastiques(k).k =✓
'0';
    end

case 10 %Inertes, solos e terras

    k = length(projecto(carproj).bdados.material.inertes_solos_terras) + 1;
    projecto(carproj).bdados.material.inertes_solos_terras(k).nome = get✓
(handles.materiais_nome, 'string');
    projecto(carproj).bdados.material.inertes_solos_terras(k).classificacao =✓
'Inertes, solos e terras';
    projecto(carproj).bdados.material.inertes_solos_terras(k).rho = str2double(✓
get(handles.materiais_rho, 'string') );
    projecto(carproj).bdados.material.inertes_solos_terras(k).k = str2double(✓
get(handles.materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.inertes_solos_terras(k).nome)
        projecto(carproj).bdados.material.inertes_solos_terras(k).nome = 'Sem✓
nome';
    end
    if isnan(projecto(carproj).bdados.material.inertes_solos_terras(k).rho)
        projecto(carproj).bdados.material.inertes_solos_terras(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.inertes_solos_terras(k).k)
        projecto(carproj).bdados.material.inertes_solos_terras(k).k = '0';
    end

case 11 %Isolamento Térmico

    k = length(projecto(carproj).bdados.material.isolamento) + 1;
    projecto(carproj).bdados.material.isolamento(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.isolamento(k).classificacao = 'Isolante✓
Térmico';
    projecto(carproj).bdados.material.isolamento(k).rho = str2double( get✓
(handles.materiais_rho, 'string') );
    projecto(carproj).bdados.material.isolamento(k).k = str2double( get(handles.✓
materiais_k, 'string') );

```

```

    if isempty(projecto(carproj).bdados.material.isolamento(k).nome)
        projecto(carproj).bdados.material.isolamento(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.isolamento(k).rho)
        projecto(carproj).bdados.material.isolamento(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.isolamento(k).k)
        projecto(carproj).bdados.material.isolamento(k).k = '0';
    end
end

case 12 %Madeiras e derivados

    k = length(projecto(carproj).bdados.material.madeira) + 1;
    projecto(carproj).bdados.material.madeira(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.madeira(k).classificacao = 'Madeira e✓
derivados';
    projecto(carproj).bdados.material.madeira(k).rho = str2double( get(handles.✓
materiais_rho, 'string') );
    projecto(carproj).bdados.material.madeira(k).k = str2double( get(handles.✓
materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.madeira(k).nome)
        projecto(carproj).bdados.material.madeira(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.madeira(k).rho)
        projecto(carproj).bdados.material.madeira(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.madeira(k).k)
        projecto(carproj).bdados.material.madeira(k).k = '0';
    end
end

case 13 %Materiais Plásticos

    k = length(projecto(carproj).bdados.material.plasticos) + 1;
    projecto(carproj).bdados.material.plasticos(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.plasticos(k).classificacao = 'Materiais✓
Plásticos';
    projecto(carproj).bdados.material.plasticos(k).rho = str2double( get✓
(handles.materiais_rho, 'string') );
    projecto(carproj).bdados.material.plasticos(k).k = str2double( get(handles.✓
materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.plasticos(k).nome)
        projecto(carproj).bdados.material.plasticos(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.plasticos(k).rho)
        projecto(carproj).bdados.material.plasticos(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.plasticos(k).k)
        projecto(carproj).bdados.material.plasticos(k).k = '0';
    end
end

case 14 %Material Cerâmico

    k = length(projecto(carproj).bdados.material.ceramico) + 1;
    projecto(carproj).bdados.material.ceramico(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.ceramico(k).classificacao = 'Material✓

```

```

Cerâmico';
    projecto(carproj).bdados.material.ceramico(k).rho = str2double( get(handles.✓
materiais_rho, 'string') );
    projecto(carproj).bdados.material.ceramico(k).k = str2double( get(handles.✓
materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.ceramico(k).nome)
        projecto(carproj).bdados.material.ceramico(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.ceramico(k).rho)
        projecto(carproj).bdados.material.ceramico(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.ceramico(k).k)
        projecto(carproj).bdados.material.ceramico(k).k = '0';
    end

case 15 %Metais

    k = length(projecto(carproj).bdados.material.metais) + 1;
    projecto(carproj).bdados.material.metais(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.metais(k).classificacao = 'Metais';
    projecto(carproj).bdados.material.metais(k).rho = str2double( get(handles.✓
materiais_rho, 'string') );
    projecto(carproj).bdados.material.metais(k).k = str2double( get(handles.✓
materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.metais(k).nome)
        projecto(carproj).bdados.material.metais(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.metais(k).rho)
        projecto(carproj).bdados.material.metais(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.metais(k).k)
        projecto(carproj).bdados.material.metais(k).k = '0';
    end

case 16 %Pedras

    k = length(projecto(carproj).bdados.material.pedras) + 1;
    projecto(carproj).bdados.material.pedras(k).nome = get(handles.✓
materiais_nome, 'string');
    projecto(carproj).bdados.material.pedras(k).classificacao = 'Pedras';
    projecto(carproj).bdados.material.pedras(k).rho = str2double( get(handles.✓
materiais_rho, 'string') );
    projecto(carproj).bdados.material.pedras(k).k = str2double( get(handles.✓
materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.pedras(k).nome)
        projecto(carproj).bdados.material.pedras(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.pedras(k).rho)
        projecto(carproj).bdados.material.pedras(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.pedras(k).k)
        projecto(carproj).bdados.material.pedras(k).k = '0';
    end

case 17 %Revestimentos de pisos ou paredes

```

```

        k = length(projecto(carproj).bdados.material.revestimentos_pisos_paredes) + 1;
    projecto(carproj).bdados.material.revestimentos_pisos_paredes(k).nome = get(handles.materiais_nome, 'string');
    projecto(carproj).bdados.material.revestimentos_pisos_paredes(k).classificacao = 'Revestimento de pisos ou de paredes';
    projecto(carproj).bdados.material.revestimentos_pisos_paredes(k).rho = str2double( get(handles.materiais_rho, 'string') );
    projecto(carproj).bdados.material.revestimentos_pisos_paredes(k).k = str2double( get(handles.materiais_k, 'string') );

    if isempty(projecto(carproj).bdados.material.revestimentos_pisos_paredes(k).nome)
        projecto(carproj).bdados.material.revestimentos_pisos_paredes(k).nome = 'Sem nome';
    end
    if isnan(projecto(carproj).bdados.material.revestimentos_pisos_paredes(k).rho)
        projecto(carproj).bdados.material.revestimentos_pisos_paredes(k).rho = '0';
    end
    if isnan( projecto(carproj).bdados.material.revestimentos_pisos_paredes(k).k)
        projecto(carproj).bdados.material.revestimentos_pisos_paredes(k).k = '0';
    end

    case 18 %Vidro

        k = length(projecto(carproj).bdados.material.vidros) + 1;
        projecto(carproj).bdados.material.vidros(k).nome = get(handles.materiais_nome, 'string');
        projecto(carproj).bdados.material.vidros(k).classificacao = 'Vidros';
        projecto(carproj).bdados.material.vidros(k).rho = str2double( get(handles.materiais_rho, 'string') );
        projecto(carproj).bdados.material.vidros(k).k = str2double( get(handles.materiais_k, 'string') );

        if isempty(projecto(carproj).bdados.material.vidros(k).nome)
            projecto(carproj).bdados.material.vidros(k).nome = 'Sem nome';
        end
        if isnan(projecto(carproj).bdados.material.vidros(k).rho)
            projecto(carproj).bdados.material.vidros(k).rho = '0';
        end
        if isnan( projecto(carproj).bdados.material.vidros(k).k)
            projecto(carproj).bdados.material.vidros(k).k = '0';
        end

    end

    save projecto.mat projecto

    warndlg('Material registado!!!', 'REGISTO');

    load projecto.mat

    w = {projecto(carproj).bdados.material.agua(1:end).nome, projecto(carproj).bdados.material.argamassa(1:end).nome, projecto(carproj).bdados.material.betao(1:end).nome, ...

```

```

    projecto(carproj).bdados.material.borrachas(1:end).nome,projecto(carproj).bdados.material.fibrocimento(1:end).nome,...
    projecto(carproj).bdados.material.gases(1:end).nome,projecto(carproj).bdados.material.gessos(1:end).nome,projecto(carproj).bdados.material.impermeabilizacao_mastiques(1:end).nome,...
    projecto(carproj).bdados.material.inertes_solos_terras(end).nome,projecto(carproj).bdados.material.isolamento(1:end).nome,...
    projecto(carproj).bdados.material.madeira(1:end).nome,projecto(carproj).bdados.material.plasticos(1:end).nome,projecto(carproj).bdados.material.ceramico(1:end).nome,...
    projecto(carproj).bdados.material.metais(1:end).nome,projecto(carproj).bdados.material.pedras(1:end).nome,projecto(carproj).bdados.material.revestimentos_pisos_paredes(1:end).nome,...
    projecto(carproj).bdados.material.vidros(1:end).nome};

set(handles.materiais_lista,'String',w)
set(handles.materiais_tipo,'value',1)
set(handles.materiais_nome,'string','')
set(handles.materiais_rho,'string','')
set(handles.materiais_k,'string','')

clearvars k w

% --- Executes on selection change in materiais_lista.
function materiais_lista_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_lista (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns materiais_lista contents as cell array
%         contents{get(hObject,'Value')} returns selected item from materiais_lista

% --- Executes during object creation, after setting all properties.
function materiais_lista_CreateFcn(hObject, eventdata, handles)
% hObject    handle to materiais_lista (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in materiais_visualizar.
function materiais_visualizar_Callback(hObject, eventdata, handles)

load projeto.mat
global carproj

switch get(handles.materiais_tipo,'value')
    case 1 %(Seleccione)

```

```

case 2 %Água
k = get(handles.materiais_lista, 'value');
set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.agua(k).✓
nome)
set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.agua(k).✓
rho)
set(handles.materiais_k, 'string', projecto(carproj).bdados.material.agua(k).k)
case 3 %Argamassa
k = get(handles.materiais_lista, 'value');
set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.argamassa✓
(k).nome)
set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.argamassa✓
(k).rho)
set(handles.materiais_k, 'string', projecto(carproj).bdados.material.argamassa(k).✓
k)
case 4 %Betão
k = get(handles.materiais_lista, 'value');
set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.betao(k).✓
nome)
set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.betao(k).✓
rho)
set(handles.materiais_k, 'string', projecto(carproj).bdados.material.betao(k).k)
case 5 %Borrachas
k = get(handles.materiais_lista, 'value');
set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.borrachas✓
(k).nome)
set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.borrachas✓
(k).rho)
set(handles.materiais_k, 'string', projecto(carproj).bdados.material.borrachas(k).✓
k)
case 6 %Fibrocimento
k = get(handles.materiais_lista, 'value');
set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.✓
fibrocimento(k).nome)
set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.✓
fibrocimento(k).rho)
set(handles.materiais_k, 'string', projecto(carproj).bdados.material.fibrocimento✓
(k).k)
case 7 %Gases
k = get(handles.materiais_lista, 'value');
set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.gases(k).✓
nome)
set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.gases(k).✓
rho)
set(handles.materiais_k, 'string', projecto(carproj).bdados.material.gases(k).k)
case 8 %Gessos
k = get(handles.materiais_lista, 'value');
set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.gessos(k).✓
nome)
set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.gessos(k).✓
rho)
set(handles.materiais_k, 'string', projecto(carproj).bdados.material.gessos(k).k)
case 9 %Impermeabilização e mastiques
k = get(handles.materiais_lista, 'value');
set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.✓
impermeabilizacao_mastiques(k).nome)
set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.✓
impermeabilizacao_mastiques(k).rho)
set(handles.materiais_k, 'string', projecto(carproj).bdados.material.✓
impermeabilizacao_mastiques(k).k)

```

```

    case 10 % Inertes, Solos e Terras
    k = get(handles.materiais_lista, 'value');
    set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.✓
inertes_solos_terras(k).nome)
    set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.✓
inertes_solos_terras(k).rho)
    set(handles.materiais_k, 'string', projecto(carproj).bdados.material.✓
inertes_solos_terras(k).k)
    case 11 %Isolamento Térmico
    k = get(handles.materiais_lista, 'value');
    set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.isolamento✓
(k).nome)
    set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.isolamento✓
(k).rho)
    set(handles.materiais_k, 'string', projecto(carproj).bdados.material.isolamento✓
(k).k)
    case 12 %Madeiras e Derivados
    k = get(handles.materiais_lista, 'value');
    set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.madeira✓
(k).nome)
    set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.madeira(k).✓
rho)
    set(handles.materiais_k, 'string', projecto(carproj).bdados.material.madeira(k).k)
    case 13 %Materiais Plásticos
    k = get(handles.materiais_lista, 'value');
    set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.plasticos✓
(k).nome)
    set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.plasticos✓
(k).rho)
    set(handles.materiais_k, 'string', projecto(carproj).bdados.material.plasticos(k).✓
k)
    case 14 %Materiais Cerâmico
    k = get(handles.materiais_lista, 'value');
    set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.ceramico✓
(k).nome)
    set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.ceramico✓
(k).rho)
    set(handles.materiais_k, 'string', projecto(carproj).bdados.material.ceramico(k).✓
k)
    case 15 %Metais
    k = get(handles.materiais_lista, 'value');
    set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.metais(k).✓
nome)
    set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.metais(k).✓
rho)
    set(handles.materiais_k, 'string', projecto(carproj).bdados.material.metais(k).k)
    case 16 %Pedras
    k = get(handles.materiais_lista, 'value');
    set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.pedras(k).✓
nome)
    set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.pedras(k).✓
rho)
    set(handles.materiais_k, 'string', projecto(carproj).bdados.material.pedras(k).k)
    case 17 %Revestimentos de pisos ou de paredes
    k = get(handles.materiais_lista, 'value');
    set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.✓
revestimentos_pisos_paredes(k).nome)
    set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.✓
revestimentos_pisos_paredes(k).rho)
    set(handles.materiais_k, 'string', projecto(carproj).bdados.material.✓

```



```

revestimentos_pisos_paredes(k).k)
    case 18 %Vidros
        k = get(handles.materiais_lista, 'value');
        set(handles.materiais_nome, 'string', projecto(carproj).bdados.material.vidros(k).✓
nome)
        set(handles.materiais_rho, 'string', projecto(carproj).bdados.material.vidros(k).✓
rho)
        set(handles.materiais_k, 'string', projecto(carproj).bdados.material.vidros(k).k)
    end

clearvars k
% --- Executes on button press in materiais_eliminar.
function materiais_eliminar_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_eliminar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

load projecto.mat
global carproj

switch get(handles.materiais_tipo, 'value')
    case 1
    case 2% Água

        k=get(handles.materiais_lista, 'value');
        if k>2

            choice = questdlg('Tem a certeza que deseja eliminar o✓
material:', 'Atenção!', 'Sim', 'Não', 'Não');

            switch choice
                case 'Sim'
                    projecto(carproj).bdados.material.agua(k)=[];
                    save projecto.mat projecto
                    f = warndlg('Material eliminado!!!', 'Eliminar');
                    uiwait(f)
                case 'Não'
            end

        else
            f = warndlg('Não é possível eliminar o material!!!', 'Erro');
            uiwait(f)
        end

    case 3% Argamassa

        k=get(handles.materiais_lista, 'value');
        if k>3

            choice = questdlg('Tem a certeza que deseja eliminar o✓
material:', 'Atenção!', 'Sim', 'Não', 'Não');

            switch choice
                case 'Sim'
                    projecto(carproj).bdados.material.argamassa(k)=[];
                    save projecto.mat projecto
                    f = warndlg('Material eliminado!!!', 'Eliminar');
                    uiwait(f)
                case 'Não'
            end

        else

```

```

        f = warndlg('Não é possível eliminar o material!!!', 'Erro');
        uiwait(f)
    end
case 4 %Betão

k=get(handles.materiais_lista,'value');
if k>18
    choice = questdlg('Tem a certeza que deseja eliminar o material:', 'Atenção!', 'Sim', 'Não', 'Não');

    switch choice
    case 'Sim'
        projecto(carproj).bdados.material.betao(k)=[];
        save projecto.mat projecto
        f = warndlg('Material eliminado!!!', 'Eliminar');
        uiwait(f)
    case 'Não'
    end

else
    f = warndlg('Não é possível eliminar o material!!!', 'Erro');
    uiwait(f)
end

case 5 %Borrachas

k=get(handles.materiais_lista,'value');
if k>7
    choice = questdlg('Tem a certeza que deseja eliminar o material:', 'Atenção!', 'Sim', 'Não', 'Não');

    switch choice
    case 'Sim'
        projecto(carproj).bdados.material.borrachas(k)=[];
        save projecto.mat projecto
        f = warndlg('Material eliminado!!!', 'Eliminar');
        uiwait(f)
    case 'Não'
    end

else
    f = warndlg('Não é possível eliminar o material!!!', 'Erro');
    uiwait(f)
end

case 6 %Fibrocimento

k=get(handles.materiais_lista,'value');
if k>2
    choice = questdlg('Tem a certeza que deseja eliminar o material:', 'Atenção!', 'Sim', 'Não', 'Não');

    switch choice
    case 'Sim'
        projecto(carproj).bdados.material.fibrocimento(k)=[];
        save projecto.mat projecto
        f = warndlg('Material eliminado!!!', 'Eliminar');
        uiwait(f)
    case 'Não'
    end

else
    f = warndlg('Não é possível eliminar o material!!!', 'Erro');
    uiwait(f)
end

```

```

end

case 7 % Gases

    k=get(handles.materiais_lista,'value');
    if k>5
        choice = questdlg('Tem a certeza que deseja eliminar o ✓
material:', 'Atenção!', 'Sim', 'Não', 'Não');

        switch choice
        case 'Sim'
            projecto(carproj).bdados.material.gases(k)=[];
            save projecto.mat projecto
            f = warndlg('Material eliminado!!!', 'Eliminar');
            uiwait(f)
        case 'Não'
        end

    else
        f = warndlg('Não é possível eliminar o material!!!', 'Erro');
        uiwait(f)
    end

case 8 % Gessos

    k=get(handles.materiais_lista,'value');
    if k>6
        choice = questdlg('Tem a certeza que deseja eliminar o ✓
material:', 'Atenção!', 'Sim', 'Não', 'Não');

        switch choice
        case 'Sim'
            projecto(carproj).bdados.material.gessos(k)=[];
            save projecto.mat projecto
            f = warndlg('Material eliminado!!!', 'Eliminar');
            uiwait(f)
        case 'Não'
        end

    else
        f = warndlg('Não é possível eliminar o material!!!', 'Erro');
        uiwait(f)
    end

case 9 %Materiais de impermeabilização e mastiques

    k=get(handles.materiais_lista,'value');
    if k>7
        choice = questdlg('Tem a certeza que deseja eliminar o ✓
material:', 'Atenção!', 'Sim', 'Não', 'Não');

        switch choice
        case 'Sim'
            projecto(carproj).bdados.material.✓
impermeabilizacao_mastiques(k)=[];
            save projecto.mat projecto
            f = warndlg('Material eliminado!!!', 'Eliminar');
            uiwait(f)
        case 'Não'
        end

    else
        f = warndlg('Não é possível eliminar o material!!!', 'Erro');
    end
end

```

```

        uiwait(f)
    end

    case 10 % Inertes, Solos e Terras

        k=get(handles.materiais_lista,'value');
        if k>3
            choice = questdlg('Tem a certeza que deseja eliminar o ✓
material:', 'Atenção!', 'Sim', 'Não', 'Não');

            switch choice
            case 'Sim'
                projecto(carproj).bdados.material.inertes_solos_terras(k)=✓
[];

                save projecto.mat projecto
                f = warndlg('Material eliminado!!!', 'Eliminar');
                uiwait(f)
            case 'Não'
            end

        else
            f = warndlg('Não é possível eliminar o material!!!', 'Erro');
            uiwait(f)
        end

    case 11 % Isolantes Térmicos

        k=get(handles.materiais_lista,'value');
        if k>13
            choice = questdlg('Tem a certeza que deseja eliminar o ✓
material:', 'Atenção!', 'Sim', 'Não', 'Não');

            switch choice
            case 'Sim'
                projecto(carproj).bdados.material.isolamento(k)=[];
                save projecto.mat projecto
                f = warndlg('Material eliminado!!!', 'Eliminar');
                uiwait(f)
            case 'Não'
            end

        else
            f = warndlg('Não é possível eliminar o material!!!', 'Erro');
            uiwait(f)
        end

    case 12 % Madeiras e derivados

        k=get(handles.materiais_lista,'value');
        if k>16
            choice = questdlg('Tem a certeza que deseja eliminar o ✓
material:', 'Atenção!', 'Sim', 'Não', 'Não');

            switch choice
            case 'Sim'
                projecto(carproj).bdados.material.madeira(k)=[];
                save projecto.mat projecto
                f = warndlg('Material eliminado!!!', 'Eliminar');
                uiwait(f)
            case 'Não'
            end

        else

```

```

f = warndlg('Não é possível eliminar o material!!!', 'Erro');
uiwait(f)
end

case 13 % Materiais Plásticos

    k=get(handles.materiais_lista,'value');
    if k>14
        choice = questdlg('Tem a certeza que deseja eliminar o material:', 'Atenção!', 'Sim', 'Não', 'Não');

        switch choice
            case 'Sim'
                projecto(carproj).bdados.material.plasticos(k)=[];
                save projecto.mat projecto
                f = warndlg('Material eliminado!!!', 'Eliminar');
                uiwait(f)
            case 'Não'
            end
        else
            f = warndlg('Não é possível eliminar o material!!!', 'Erro');
            uiwait(f)
        end

case 14 % Material Cerâmico

    k=get(handles.materiais_lista,'value');
    if k>1
        choice = questdlg('Tem a certeza que deseja eliminar o material:', 'Atenção!', 'Sim', 'Não', 'Não');

        switch choice
            case 'Sim'
                projecto(carproj).bdados.material.ceramico(k)=[];
                save projecto.mat projecto
                f = warndlg('Material eliminado!!!', 'Eliminar');
                uiwait(f)
            case 'Não'
            end
        else
            f = warndlg('Não é possível eliminar o material!!!', 'Erro');
            uiwait(f)
        end

case 15 % Metais

    k=get(handles.materiais_lista,'value');
    if k>11
        choice = questdlg('Tem a certeza que deseja eliminar o material:', 'Atenção!', 'Sim', 'Não', 'Não');

        switch choice
            case 'Sim'
                projecto(carproj).bdados.material.metais(k)=[];
                save projecto.mat projecto
                f = warndlg('Material eliminado!!!', 'Eliminar');
                uiwait(f)
            case 'Não'
            end
        end
    end

```

```

else
    f = warndlg('Não é possível eliminar o material!!!', 'Erro');
    uiwait(f)
end

case 16 % Pedras

    k=get(handles.materiais_lista,'value');
    if k>18
        choice = questdlg('Tem a certeza que deseja eliminar o material:', 'Atenção!', 'Sim', 'Não', 'Não');

        switch choice
            case 'Sim'
                projecto(carproj).bdados.material.pedras(k)=[];
                save projecto.mat projecto
                f = warndlg('Material eliminado!!!', 'Eliminar');
                uiwait(f)
            case 'Não'
            end

        else
            f = warndlg('Não é possível eliminar o material!!!', 'Erro');
            uiwait(f)
        end

    case 17 % Revestimentos de pisos ou de paredes

        k=get(handles.materiais_lista,'value');
        if k>10
            choice = questdlg('Tem a certeza que deseja eliminar o material:', 'Atenção!', 'Sim', 'Não', 'Não');

            switch choice
                case 'Sim'
                    projecto(carproj).bdados.material.revestimentos_pisos_paredes(k)=[];
                    save projecto.mat projecto
                    f = warndlg('Material eliminado!!!', 'Eliminar');
                    uiwait(f)
                case 'Não'
                end

            else
                f = warndlg('Não é possível eliminar o material!!!', 'Erro');
                uiwait(f)
            end

        case 18 % Vidro

            k=get(handles.materiais_lista,'value');
            if k>3
                choice = questdlg('Tem a certeza que deseja eliminar o material:', 'Atenção!', 'Sim', 'Não', 'Não');

                switch choice
                    case 'Sim'
                        projecto(carproj).bdados.material.vidros(k)=[];
                        save projecto.mat projecto
                        f = warndlg('Material eliminado!!!', 'Eliminar');
                        uiwait(f)
                    case 'Não'
                    end

                else
                    f = warndlg('Não é possível eliminar o material!!!', 'Erro');
                    uiwait(f)
                end

            end
        end
    end
end

```

```

        case 'Não'
        end

    else
        f = warndlg('Não é possível eliminar o material!!!', 'Erro');
        uiwait(f)
    end

end

load projecto.mat

w = {projecto(carproj).bdados.material.agua(1:end).nome,projecto(carproj).bdados.✓
material.argamassa(1:end).nome,projecto(carproj).bdados.material.betao(1:end).✓
nome,...
    projecto(carproj).bdados.material.borrachas(1:end).nome,projecto(carproj).✓
bdados.material.fibrocimento(1:end).nome,...
    projecto(carproj).bdados.material.gases(1:end).nome,projecto(carproj).bdados.✓
material.gessos(1:end).nome,projecto(carproj).bdados.material.✓
impermeabilizacao_mastiques(1:end).nome,...
    projecto(carproj).bdados.material.inertes_solos_terras(end).nome,projecto✓
(carproj).bdados.material.isolamento(1:end).nome,...
    projecto(carproj).bdados.material.madeira(1:end).nome,projecto(carproj).bdados.✓
material.plasticos(1:end).nome,projecto(carproj).bdados.material.ceramico(1:end).✓
nome,...
    projecto(carproj).bdados.material.metais(1:end).nome,projecto(carproj).bdados.✓
material.pedras(1:end).nome,projecto(carproj).bdados.material.✓
revestimentos_pisos_paredes(1:end).nome,...
    projecto(carproj).bdados.material.vidros(1:end).nome};

set(handles.materiais_lista,'String',w)
set(handles.materiais_tipo,'value',1)
set(handles.materiais_nome,'string','')
set(handles.materiais_rho,'string','')
set(handles.materiais_k,'string','')

clearvars k f w choice

% -----
function materiais_inicio_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_inicio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('inicio.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_local_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_local (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('local.m')

close(findobj('type','figure','name','Materiais'))

```

```

% -----
function materiais_elementos_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_elementos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function materiais_horarios_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_horarios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function materiais_relatorios_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_relatorios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('vidro.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_caixilho_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('caixilho.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_adiporta_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_adiporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('Porta.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_verporta_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_verporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```



```

% handles      structure with handles and user data (see GUIDATA)

run('porta_vereliminar.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_jansimples_Callback(hObject, eventdata, handles)
% hObject      handle to materiais_jansimples (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('janela_simples.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_jandupla_Callback(hObject, eventdata, handles)
% hObject      handle to materiais_jandupla (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('janela_dupla.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_duplafolha_Callback(hObject, eventdata, handles)
% hObject      handle to materiais_duplafolha (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_verjanela_Callback(hObject, eventdata, handles)
% hObject      handle to materiais_verjanela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('janela_vereliminar.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_adhorario_Callback(hObject, eventdata, handles)
% hObject      handle to materiais_adhorario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('definir_horario.m')

close(findobj('type','figure','name','Materiais'))

```

```

% -----
function materiais_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_defhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_defespaco_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_defespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('definir_espaco.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('ver_espaco.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_adparede_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_adparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('adicionarparede.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_verparede_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_verparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('ver_parede.m')

close(findobj('type','figure','name','Materiais'))

% -----
function materiais_materiais_Callback(hObject, eventdata, handles)
% hObject    handle to materiais_materiais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```
% -----  
function materiais_espaco_Callback(hObject, eventdata, handles)  
% hObject    handle to materiais_espaco (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% -----  
function materiais_janela_Callback(hObject, eventdata, handles)  
% hObject    handle to materiais_janela (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% -----  
function materiais_porta_Callback(hObject, eventdata, handles)  
% hObject    handle to materiais_porta (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% -----  
function materiais_parede_Callback(hObject, eventdata, handles)  
% hObject    handle to materiais_parede (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

```

function varargout = relatorios(varargin)
% RELATORIOS_VERPORTA MATLAB code for relatorios_verporta.fig
%     RELATORIOS_VERPORTA, by itself, creates a new RELATORIOS_VERPORTA or raises
the existing
%     singleton*.
%
%     H = RELATORIOS_VERPORTA returns the handle to a new RELATORIOS_VERPORTA or
the handle to
%     the existing singleton*.
%
%     RELATORIOS_VERPORTA('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in RELATORIOS_VERPORTA.M with the given input
arguments.
%
%     RELATORIOS_VERPORTA('Property','Value',...) creates a new RELATORIOS_VERPORTA
or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before relatorios_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to relatorios_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help relatorios_verporta

% Last Modified by GUIDE v2.5 21-Jul-2016 20:49:02

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @relatorios_OpeningFcn, ...
                  'gui_OutputFcn',  @relatorios_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before relatorios_verporta is made visible.
function relatorios_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to relatorios_verporta (see VARARGIN)

% Choose default command line output for relatorios_verporta
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

movegui(gcf, 'center')

set(handles.figure1, 'Name', 'Relatórios')

load projecto.mat
global carproj

if isempty(projecto(carproj).espaco)==1

    set(handles.relatorios_espaco, 'String', 'Não existem espaços definidos')
else
set(handles.relatorios_espaco, 'String', {projecto(carproj).espaco.nome})
end

% UIWAIT makes relatorios_verporta wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = relatorios_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% -----
function relatorios_inicio_Callback(hObject, eventdata, handles)
% hObject      handle to relatorios_inicio (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('inicio.m')

close(findobj('type','figure','name','Relatórios'))

% -----
function relatorios_checklocal_Callback(hObject, eventdata, handles)
% hObject      handle to relatorios_checklocal (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function relatorios_material_Callback(hObject, eventdata, handles)
% hObject      handle to relatorios_material (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('registamateriais.m')

```

```

close(findobj('type','figure','name','Relatórios'))

% -----
function relatorios_adhorario_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_adhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_horario.m')

close(findobj('type','figure','name','Relatórios'))

% -----
function relatorios_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_defhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Relatórios'))

% -----
function relatorios_defespco_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_defespco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

close(findobj('type','figure','name','Relatórios'))

% -----
function relatorios_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('ver_espaco.m')

close(findobj('type','figure','name','Relatórios'))

% -----
function relatorios_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('vidro.m')

close(findobj('type','figure','name','Relatórios'))

% -----
function relatorios_caixilho_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('caixilho.m')

```

```
close(findobj('type','figure','name','Relatórios'))
```

```
% -----  
function relatorios_adparede_Callback(hObject, eventdata, handles)  
% hObject    handle to relatorios_adparede (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('adicionarparede.m')
```

```
close(findobj('type','figure','name','Relatórios'))
```

```
% -----  
function relatorios_verparede_Callback(hObject, eventdata, handles)  
% hObject    handle to relatorios_verparede (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('ver_parede.m')
```

```
close(findobj('type','figure','name','Relatórios'))
```

```
% -----  
function relatorios_adporta_Callback(hObject, eventdata, handles)  
% hObject    handle to relatorios_adporta (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('Porta.m')
```

```
close(findobj('type','figure','name','Relatórios'))
```

```
% -----  
function relatorios_verporta_Callback(hObject, eventdata, handles)  
% hObject    handle to relatorios_verporta (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('ver_porta.m')
```

```
close(findobj('type','figure','name','Relatórios'))
```

```
% -----  
function relatorios_simples_Callback(hObject, eventdata, handles)  
% hObject    handle to relatorios_simples (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('janela_simples.m')
```

```
close(findobj('type','figure','name','Relatórios'))
```

```
% -----  
function relatorios_dupla_Callback(hObject, eventdata, handles)  
% hObject    handle to relatorios_dupla (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('janela_dupla.m')
```

```

close(findobj('type','figure','name','Relatórios'))

% -----
function relatorios_duplafolha_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_duplafolha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Relatórios'))

% -----
function relatorios_verjanela_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_verjanela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')

close(findobj('type','figure','name','Relatórios'))

% --- Executes on button press in relatorios_checkmateriais.
function relatorios_checkmateriais_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_checkmateriais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of relatorios_checkmateriais

% --- Executes on button press in relatorios_checkvidros.
function relatorios_checkvidros_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_checkvidros (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of relatorios_checkvidros

% --- Executes on button press in relatorios_checkcaixilhos.
function relatorios_checkcaixilhos_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_checkcaixilhos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of relatorios_checkcaixilhos

% --- Executes on button press in relatorios_checkjanelas.
function relatorios_checkjanelas_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_checkjanelas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of relatorios_checkjanelas

% --- Executes on button press in relatorios_checkportas.

```



```

function relatorios_checkportas_Callback(hObject, eventdata, handles)
% hObject      handle to relatorios_checkportas (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of relatorios_checkportas

% --- Executes on button press in relatorios_checkparedes.
function relatorios_checkparedes_Callback(hObject, eventdata, handles)
% hObject      handle to relatorios_checkparedes (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of relatorios_checkparedes

% --- Executes on button press in relatorios_checkespacos.
function relatorios_checkespacos_Callback(hObject, eventdata, handles)
% hObject      handle to relatorios_checkespacos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of relatorios_checkespacos

% --- Executes on button press in relatorios_checkhorarios.
function relatorios_checkhorarios_Callback(hObject, eventdata, handles)
% hObject      handle to relatorios_checkhorarios (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of relatorios_checkhorarios

% --- Executes on button press in relatorios_gerar.
function relatorios_gerar_Callback(hObject, eventdata, handles)
% hObject      handle to relatorios_gerar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

load projecto.mat

global carproj

filename = 'Relatórios.xlsx';

filename_template = 'resultados_template.xlsx';

copyfile(filename_template,filename)

xlswrite(filename,{projecto(carproj).nome},'Projeto','E12')
xlswrite(filename,{projecto(carproj).instituicao},'Projeto','E14')
xlswrite(filename,{projecto(carproj).autor},'Projeto','E16')

% Local
h=waitbar(0,'A criar relatórios');
if get(handles.relatorios_checklocal,'Value')==1

```

```

A = {projecto(carproj).local.local;projecto(carproj).local.latitude;projecto
(carproj).local.longitude;projecto(carproj).local.altitude;...
    projecto(carproj).local.tempDBV; projecto(carproj).local.tempWBV; projecto
(carproj).local.tempDTV; projecto(carproj).local.tempDBI;...
    projecto(carproj).local.tempWBI; projecto(carproj).local.nome_ficheiro};

xlswrite(filename,A,'Local','B1')

clearvars A
end
close(h)
% Materiais
h=waitbar(0.1,'A criar relatórios');
if get(handles.relatorios_checkmateriais,'Value')==1

    A = {projecto(carproj).bdados.material.agua.nome,projecto(carproj).bdados.
material.argamassa.nome,projecto(carproj).bdados.material.betao.nome,...
        projecto(carproj).bdados.material.borrachas.nome,projecto(carproj).bdados.
material.fibrocimento.nome,projecto(carproj).bdados.material.gases.nome,...
        projecto(carproj).bdados.material.gessos.nome,projecto(carproj).bdados.
material.impermeabilizacao_mastiques.nome,...
        projecto(carproj).bdados.material.inertes_solos_terras.nome,projecto
(carproj).bdados.material.isolamento.nome,...
        projecto(carproj).bdados.material.madeira.nome,projecto(carproj).bdados.
material.plasticos.nome,projecto(carproj).bdados.material.ceramico.nome,...
        projecto(carproj).bdados.material.metais.nome,projecto(carproj).bdados.
material.pedras.nome,...
        projecto(carproj).bdados.material.revestimentos_pisos_paredes.nome,projecto
(carproj).bdados.material.vidros.nome}';

    B = {projecto(carproj).bdados.material.agua.classificacao,projecto(carproj).
bdados.material.argamassa.classificacao,projecto(carproj).bdados.material.betao.
classificacao,...
        projecto(carproj).bdados.material.borrachas.classificacao,projecto(carproj).
bdados.material.fibrocimento.classificacao,projecto(carproj).bdados.material.gases.
classificacao,...
        projecto(carproj).bdados.material.gessos.classificacao,projecto(carproj).
bdados.material.impermeabilizacao_mastiques.classificacao,...
        projecto(carproj).bdados.material.inertes_solos_terras.classificacao,
projecto(carproj).bdados.material.isolamento.classificacao,...
        projecto(carproj).bdados.material.madeira.classificacao,projecto(carproj).
bdados.material.plasticos.classificacao,projecto(carproj).bdados.material.ceramico.
classificacao,...
        projecto(carproj).bdados.material.metais.classificacao,projecto(carproj).
bdados.material.pedras.classificacao,...
        projecto(carproj).bdados.material.revestimentos_pisos_paredes.classificacao,
projecto(carproj).bdados.material.vidros.classificacao}';

    C = {projecto(carproj).bdados.material.agua.rho,projecto(carproj).bdados.
material.argamassa.rho,projecto(carproj).bdados.material.betao.rho,...
        projecto(carproj).bdados.material.borrachas.rho,projecto(carproj).bdados.
material.fibrocimento.rho,projecto(carproj).bdados.material.gases.rho,...
        projecto(carproj).bdados.material.gessos.rho,projecto(carproj).bdados.
material.impermeabilizacao_mastiques.rho,...
        projecto(carproj).bdados.material.inertes_solos_terras.rho,projecto
(carproj).bdados.material.isolamento.rho,...
        projecto(carproj).bdados.material.madeira.rho,projecto(carproj).bdados.
material.plasticos.rho,projecto(carproj).bdados.material.ceramico.rho,...

```

```

        projecto(carproj).bdados.material.metalis.rho,projecto(carproj).bdados.✓
material.pedras.rho,...
        projecto(carproj).bdados.material.revestimentos_pisos_paredes.rho,projecto✓
(carproj).bdados.material.vidros.rho}';

D = {projecto(carproj).bdados.material.agua.k,projecto(carproj).bdados.✓
material.argamassa.k,projecto(carproj).bdados.material.betao.k,...
        projecto(carproj).bdados.material.borrachas.k,projecto(carproj).bdados.✓
material.fibrocimento.k,projecto(carproj).bdados.material.gases.k,...
        projecto(carproj).bdados.material.gessos.k,projecto(carproj).bdados.✓
material.impermeabilizacao_mastiques.k,...
        projecto(carproj).bdados.material.inertes_solos_terras.k,projecto(carproj).✓
bdados.material.isolamento.k,...
        projecto(carproj).bdados.material.madeira.k,projecto(carproj).bdados.✓
material.plasticos.k,projecto(carproj).bdados.material.ceramico.k,...
        projecto(carproj).bdados.material.metalis.k,projecto(carproj).bdados.✓
material.pedras.k,...
        projecto(carproj).bdados.material.revestimentos_pisos_paredes.k,projecto✓
(carproj).bdados.material.vidros.k}';

xslswrite(filename,A,'Materiais','A2')
xslswrite(filename,B,'Materiais','B2')
xslswrite(filename,C,'Materiais','C2')
xslswrite(filename,D,'Materiais','D2')

clearvars A B C D

end
close(h)
% Vidros
h=waitbar(0.2,'A criar relatórios');
if get(handles.relatorios_checkvidros,'Value')

A = {projecto(carproj).bdados.vidro.nome}';

B = {projecto(carproj).bdados.vidro.tipo}';

C = {projecto(carproj).bdados.vidro.ug}';

xslswrite(filename,A,'Vidros','A2')
xslswrite(filename,B,'vidros','B2')
xslswrite(filename,C,'Vidros','C2')

clearvars A B C
end
close(h)
%Caixilhos
h=waitbar(0.3,'A criar relatórios');
if get(handles.relatorios_checkcaixilhos,'Value')==1;

A = {projecto(carproj).bdados.caixilho.nome}';

B = {projecto(carproj).bdados.caixilho.uf}';

```

```

        xlswrite(filename,A, 'Caixilhos', 'A2')
        xlswrite(filename,B, 'Caixilhos', 'B2')

        clearvars A B

    end
    close(h)
    % Janelas
    h=waitbar(0.4, 'A criar relatórios');
    if get(handles.relatorios_checkjanelas, 'Value')==1;

        A = {projecto(carproj).bdados.janela.nome}';

        B = {projecto(carproj).bdados.janela.tipo}';

        C = {projecto(carproj).bdados.janela.uw}';

        xlswrite(filename,A, 'Janelas', 'A2')
        xlswrite(filename,B, 'Janelas', 'B2')
        xlswrite(filename,C, 'Janelas', 'C2')

        clearvars A B C
    end
    close(h)
    % Portas
    h=waitbar(0.5, 'A criar relatórios');
    if get(handles.relatorios_checkportas, 'Value')==1;

        A = {projecto(carproj).bdados.porta.nome}';

        B = {projecto(carproj).bdados.porta.ud}';

        xlswrite(filename,A, 'Portas', 'A2')
        xlswrite(filename,B, 'Portas', 'B2')

        clearvars A B

    end
    close(h)
    % Paredes, tetos, coberturas e pavimentos
    h=waitbar(0.6, 'A criar relatórios');
    if get(handles.relatorios_checkparedes, 'Value')==1;

        if isempty(projecto(carproj).bdados.elementos)==1

        else

            A = {projecto(carproj).bdados.elementos.nome}';

            B = {projecto(carproj).bdados.elementos.tipo}';

            C = {projecto(carproj).bdados.elementos.u}';

```

```

    xlswrite(filename,A,'Elementos construtivos','A2')
    xlswrite(filename,B,'Elementos construtivos','B2')
    xlswrite(filename,C,'Elementos construtivos','C2')

end

clearvars A B C

end
close(h)
h=waitbar(0.7,'A criar relatórios');
%Projectos
if get(handles.checkbox_sim,'Value')==1

    y=hora_projecto();

    diasemana = get(handles.relatorio_diasemana,'Value');

    x= analise_hora(diasemana);

k = get(handles.relatorios_espaco,'Value');

A = projecto(carproj).relatorios(k).tabela;

A.Properties.VariableNames =
{'Dia','Hora','Dia_da_semana_ou_Feriado','Horario_de_Ocupacao','Horario_de_iluminacao',
'Horario_de_equipamentos',...
'Horario_de_cargas_diversas','Carga_ocupacao_sensivel_W','Carga_ocupacao_latente_W',
'Carga_iluminacao_W','Carga_equipamentos_W',...
'Carga_diversa_sensivel_W','Carga_diversa_latente_W','Carga_infiltracoes_sensivel_W',
'Carga_infiltracoes_total_W','Carga_infiltracoes_latente_W',...
'Carga_envolvente_externa_W','Carga_envolvente_interna_W','Carga_sensivel_de_arrefecimento_W',
'Carga_latente_de_arrefecimento_W','Carga_total_de_arrefecimento_W',...
'Carga_sensivel_de_aquecimento_W','Carga_latente_de_aquecimento_W','Carga_total_de_aquecimento_W'};

writetable(A,filename,'sheet','Análise Anual','Range','A1')

B = projecto(carproj).relatorios_proj(k).tabela2;

B.Properties.VariableNames =
{'Carga_envolvente_externa_W','Carga_envolvente_interna_W','Carga_ocupacao_sensivel_W',
'Carga_ocupacao_latente_W',...
'Carga_diversa_sensivel_W','Carga_diversa_latente_W','Carga_iluminacao_W','Carga_equipamentos_W',
'Carga_infiltracoes_sensivel_W',...
'Carga_infiltracoes_total_W','Carga_infiltracoes_latente_W','Carga_sensivel_de_arrefecimento_W',
'Carga_latente_de_arrefecimento_W',...
'Carga_total_de_arrefecimento_W','Carga_envolvente_externa_inverno_W','Carga_infiltracoes_sensivel_inverno_W',
'Carga_infiltracoes_total_inverno_W',...
'Carga_infiltracoes_latente_inverno_W','Carga_sensivel_de_aquecimento_W','Carga_latente_de_aquecimento_W'};

```

```

writetable(B,filename,'sheet','Análise de projeto','Range','A1')

clearvars x y A k B
end
close(h)
h=waitbar(1,'A carregar ficheiro');
if get(handles.relatorios_simldia,'Value')==1

    % Escolher um dia para mostrar a simulação
    diab = str2double(get(handles.relatorio_dia,'String'));

    mesb = get(handles.relatorio_mes,'Value');

    y=analise_dia(mesb,diab);

    k =get(handles.relatorios_espaco,'Value');

    projecto(carproj).relatorio_dia(k).tabela(:,:)=projecto(carproj).relatorios(k).tabela((y-23):y,:);

    A = projecto(carproj).relatorio_dia(k).tabela;
    A.Properties.VariableNames =
    {'Dia','Hora','Dia_da_semana_ou_Feriado','Horario_de_Ocupacao','Horario_de_iluminacao','Horario_de_equipamentos',...
    'Horario_de_cargas_diversas','Carga_ocupacao_sensivel_W','Carga_ocupacao_latente_W','Carga_iluminacao_W','Carga_equipamentos_W',...
    'Carga_diversa_sensivel_W','Carga_diversa_latente_W','Carga_infiltracoes_sensivel_W','Carga_infiltracoes_total_W','Carga_infiltracoes_latente_W',...
    'Carga_envolvente_externa_W','Carga_envolvente_interna_W','Carga_sensivel_de_arrefecimento_W','Carga_latente_de_arrefecimento_W','Carga_total_de_arrefecimento_W',...
    'Carga_sensivel_de_aquecimento_W','Carga_latente_de_aquecimento_W','Carga_total_de_aquecimento_W'};

writetable(A,filename,'sheet','Análise dia','Range','A1')

save projecto.mat projecto

end
close(h)

warndlg('Relatórios concluídos','Aviso !')

% -----
function relatorios_local_Callback(hObject, eventdata, handles)
% hObject      handle to relatorios_local (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

run('local.m')

```

```
close(findobj('type','figure','name','Relatórios'))
```

```
% --- Executes on selection change in relatorio_diasemana.
```

```
function relatorio_diasemana_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to relatorio_diasemana (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: contents = cellstr(get(hObject,'String')) returns relatorio_diasemana as cell array
```

```
%          contents{get(hObject,'Value')} returns selected item from relatorio_diasemana
```

```
% --- Executes during object creation, after setting all properties.
```

```
function relatorio_diasemana_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to relatorio_diasemana (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: popupmenu controls usually have a white background on Windows.
```

```
%          See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
```

```
(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
% --- Executes on button press in relatorios_simldia.
```

```
function relatorios_simldia_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to relatorios_simldia (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of relatorios_simldia
```

```
function relatorio_dia_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to relatorio_dia (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of relatorio_dia as text
```

```
%          str2double(get(hObject,'String')) returns contents of relatorio_dia as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function relatorio_dia_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to relatorio_dia (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%          See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
```

```
(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

end

```
% --- Executes on selection change in relatorio_mes.
function relatorio_mes_Callback(hObject, eventdata, handles)
% hObject    handle to relatorio_mes (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: contents = cellstr(get(hObject,'String')) returns relatorio_mes contents as ✓
cell array
%         contents{get(hObject,'Value')} returns selected item from relatorio_mes
```

```
% --- Executes during object creation, after setting all properties.
function relatorio_mes_CreateFcn(hObject, eventdata, handles)
% hObject    handle to relatorio_mes (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in relatorios_simanual.
function relatorios_simanual_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_simanual (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of relatorios_simanual
```

```
% --- Executes on button press in relatorios_simmensal.
function relatorios_simmensal_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_simmensal (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of relatorios_simmensal
```

```
% --- Executes on selection change in relatorios_espaco.
function relatorios_espaco_Callback(hObject, eventdata, handles)
% hObject    handle to relatorios_espaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: contents = cellstr(get(hObject,'String')) returns relatorios_espaco ✓
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from relatorios_espaco
```

```
% --- Executes during object creation, after setting all properties.
function relatorios_espaco_CreateFcn(hObject, eventdata, handles)
% hObject    handle to relatorios_espaco (see GCBO)
```



```
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% -----
function relatorios_elementos_Callback(hObject, eventdata, handles)
% hObject handle to relatorios_elementos (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function relatorios_esp_Callback(hObject, eventdata, handles)
% hObject handle to relatorios_esp (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function relatorios_horarios_Callback(hObject, eventdata, handles)
% hObject handle to relatorios_horarios (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function relatorios_relatorios_Callback(hObject, eventdata, handles)
% hObject handle to relatorios_relatorios (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function relatorios_janela_Callback(hObject, eventdata, handles)
% hObject handle to relatorios_janela (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function relatorios_porta_Callback(hObject, eventdata, handles)
% hObject handle to relatorios_porta (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% -----
function relatorios_parede_Callback(hObject, eventdata, handles)
% hObject handle to relatorios_parede (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% --- Executes on button press in checkbox_sim.  
function checkbox_sim_Callback(hObject, eventdata, handles)  
% hObject      handle to checkbox_sim (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
  
% Hint: get(hObject,'Value') returns toggle state of checkbox_sim
```

```

function varargout = ver_espaco(varargin)
% VER_ESPACO MATLAB code for ver_espaco.fig
%     VER_ESPACO, by itself, creates a new VER_ESPACO or raises the existing
%     singleton*.
%
%     H = VER_ESPACO returns the handle to a new VER_ESPACO or the handle to
%     the existing singleton*.
%
%     VER_ESPACO('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in VER_ESPACO.M with the given input arguments.
%
%     VER_ESPACO('Property','Value',...) creates a new VER_ESPACO or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before ver_espaco_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to ver_espaco_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help ver_espaco

% Last Modified by GUIDE v2.5 22-Jul-2016 05:22:40

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @ver_espaco_OpeningFcn, ...
                  'gui_OutputFcn',  @ver_espaco_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before ver_espaco is made visible.
function ver_espaco_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to ver_espaco (see VARARGIN)

% Choose default command line output for ver_espaco
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```

```

movegui(gcf,'center')

set(handles.figure1,'Name','Ver / Eliminar Espaço')

load projecto.mat

global carproj

if isempty(projecto(carproj).espaco)

    set(handles.verespaco_lista,'String','(Não existem espaços)')
else
    set(handles.verespaco_lista,'String',{projecto(carproj).espaco.nome})
end

set(handles.verespaco_ganhosinternos,'Data',[])
set(handles.verespaco_ganhosexternos,'Data',[])

% UIWAIT makes ver_espaco wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = ver_espaco_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in verespaco_lista.
function verespaco_lista_Callback(hObject, eventdata, handles)
% hObject handle to verespaco_lista (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns verespaco_lista contents ✓
as cell array
% contents{get(hObject,'Value')} returns selected item from verespaco_lista

% --- Executes during object creation, after setting all properties.
function verespaco_lista_CreateFcn(hObject, eventdata, handles)
% hObject handle to verespaco_lista (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in verespaco_ver.
function verespaco_ver_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_ver (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

load projecto.mat

global carproj

k = get(handles.verespaco_lista, 'Value');

set(handles.verespaco_nome, 'String', projecto(carproj).espaco(k).nome)
set(handles.verespaco_temp, 'String', projecto(carproj).espaco(k).temp_interior)
set(handles.verespaco_hum, 'String', projecto(carproj).espaco(k).hum_interior)
set(handles.verespaco_ocupantes, 'String', projecto(carproj).espaco(k).ocupacao.quant)
set(handles.verespaco_ocusensivel, 'String', projecto(carproj).espaco(k).ocupacao.✓
csensivel)
set(handles.verespaco_oculatente, 'String', projecto(carproj).espaco(k).ocupacao.✓
clatente)
set(handles.verespaco_ilumpot, 'String', projecto(carproj).espaco(k).ilumina)
set(handles.verespaco_egelepot, 'String', projecto(carproj).espaco(k).eletricos)
set(handles.verespaco_divsensivel, 'String', projecto(carproj).espaco(k).diversas.✓
sensivel)
set(handles.verespaco_divlatente, 'String', projecto(carproj).espaco(k).diversas.✓
latente)
set(handles.verespaco_caudalar, 'String', projecto(carproj).espaco(k).infiltracoes.✓
caudal)
set(handles.verespaco_hrextver, 'String', projecto(carproj).espaco(k).infiltracoes.✓
HR_extv)
set(handles.verespaco_hrextinv, 'String', projecto(carproj).espaco(k).infiltracoes.✓
HR_exti)

A(:,1) = {projecto(carproj).espaco(k).ganhos_ext.tipo};
A(:,2) = {projecto(carproj).espaco(k).ganhos_ext.nome};
A(:,3) = {projecto(carproj).espaco(k).ganhos_ext.quant};
A(:,4) = {projecto(carproj).espaco(k).ganhos_ext.area};

set(handles.verespaco_ganhosexternos, 'Data', A)

B(:,1) = {projecto(carproj).espaco(k).ganhos_int.tipo};
B(:,2) = {projecto(carproj).espaco(k).ganhos_int.nome};
B(:,3) = {projecto(carproj).espaco(k).ganhos_int.quant};
B(:,4) = {projecto(carproj).espaco(k).ganhos_int.area};
B(:,5) = {projecto(carproj).espaco(k).ganhos_int.temp};

set(handles.verespaco_ganhosinternos, 'Data', B)

clearvars k A B

% --- Executes on button press in verespaco_eliminar.
function verespaco_eliminar_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_eliminar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

choice = questdlg('Deseja eliminar o espaço?', 'Eliminar✓
Espaço', 'Sim', 'Não', 'Cancelar', 'Cancelar');

```

```

switch choice

    case 'Sim'
        load projecto.mat projecto
        global carproj

        k = get(handles.verespaco_lista, 'Value');

        projecto(carproj).espaco(k)=[];

        save projecto.mat projecto

        set(handles.verespaco_lista, 'Value', 1)

        if isempty(projecto(carproj).espaco)==1
            set(handles.verespaco_lista, 'String', '(Não existem espaços)')
        else
            set(handles.verespaco_lista, 'String', {projecto(carproj).espaco.nome})
        end

        warndlg('Espaço eliminado', 'Aviso !')
    case 'Não'
    case 'Cancelar'
end

clearvars choice k

function verespaco_nome_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_nome (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of verespaco_nome as text
%         str2double(get(hObject, 'String')) returns contents of verespaco_nome as a
double

% --- Executes during object creation, after setting all properties.
function verespaco_nome_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verespaco_nome (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function verespaco_temp_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of verespaco_temp as text

```

```

%         str2double(get(hObject,'String')) returns contents of verespaco_temp as a double ✓

% --- Executes during object creation, after setting all properties.
function verespaco_temp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verespaco_temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verespaco_hum_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_hum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_hum as text
%         str2double(get(hObject,'String')) returns contents of verespaco_hum as a double ✓

% --- Executes during object creation, after setting all properties.
function verespaco_hum_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verespaco_hum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verespaco_ilumpot_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_ilumpot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_ilumpot as text
%         str2double(get(hObject,'String')) returns contents of verespaco_ilumpot as a double ✓

% --- Executes during object creation, after setting all properties.
function verespaco_ilumpot_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verespaco_ilumpot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verespaco_divlatente_Callback(hObject, eventdata, handles)
% hObject     handle to verespaco_divlatente (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_divlatente as text
%         str2double(get(hObject,'String')) returns contents of verespaco_divlatente
as a double

% --- Executes during object creation, after setting all properties.
function verespaco_divlatente_CreateFcn(hObject, eventdata, handles)
% hObject     handle to verespaco_divlatente (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verespaco_divsensivel_Callback(hObject, eventdata, handles)
% hObject     handle to verespaco_divsensivel (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_divsensivel as text
%         str2double(get(hObject,'String')) returns contents of verespaco_divsensivel
as a double

% --- Executes during object creation, after setting all properties.
function verespaco_divsensivel_CreateFcn(hObject, eventdata, handles)
% hObject     handle to verespaco_divsensivel (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verespaco_equelepot_Callback(hObject, eventdata, handles)
% hObject     handle to verespaco_equelepot (see GCBO)

```



```

% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_equelepot as text
%         str2double(get(hObject,'String')) returns contents of verespaco_equelepot as a double ✓

% --- Executes during object creation, after setting all properties.
function verespaco_equelepot_CreateFcn(hObject, eventdata, handles)
% hObject      handle to verespaco_equelepot (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verespaco_oculatente_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_oculatente (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_oculatente as text
%         str2double(get(hObject,'String')) returns contents of verespaco_oculatente ✓
as a double

% --- Executes during object creation, after setting all properties.
function verespaco_oculatente_CreateFcn(hObject, eventdata, handles)
% hObject      handle to verespaco_oculatente (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verespaco_ocusensivel_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_ocusensivel (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_ocusensivel as text
%         str2double(get(hObject,'String')) returns contents of verespaco_ocusensivel ✓
as a double

% --- Executes during object creation, after setting all properties.
function verespaco_ocusensivel_CreateFcn(hObject, eventdata, handles)
% hObject      handle to verespaco_ocusensivel (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verespaco_ocupantes_Callback(hObject, eventdata, handles)
% hObject handle to verespaco_ocupantes (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_ocupantes as text
% str2double(get(hObject,'String')) returns contents of verespaco_ocupantes
as a double

% --- Executes during object creation, after setting all properties.
function verespaco_ocupantes_CreateFcn(hObject, eventdata, handles)
% hObject handle to verespaco_ocupantes (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function verespaco_inicio_Callback(hObject, eventdata, handles)
% hObject handle to verespaco_inicio (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('inicio.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_local_Callback(hObject, eventdata, handles)
% hObject handle to verespaco_local (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('local.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_materiais_Callback(hObject, eventdata, handles)
% hObject handle to verespaco_materiais (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
run('registamateriais.m')

```

```

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_adhorario_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_adhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_horario.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_defhorario_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_defhorario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('horario.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_defespaco_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_defespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('definir_espaco.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('vidro.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_caixilho_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('caixilho.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_parede_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_parede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('adicionarparede.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----

```

```

function verespaco_verparede_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_verparede (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('ver_parede.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_adporta_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_adporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('Porta.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_verporta_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_verporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('porta_vereliminar.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_simples_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_simples (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_simples.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_dupla_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_dupla (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janeladupla.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_duplafolha_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_duplafolha (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_verjanela_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_verjanela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')

```

```

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

% -----
function verespaco_relatorios_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_relatorios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('relatorios.m')

close(findobj('type','figure','name','Ver / Eliminar Espaço'))

function verespaco_hrextinv_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_hrextinv (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_hrextinv as text
%        str2double(get(hObject,'String')) returns contents of verespaco_hrextinv as
a double

% --- Executes during object creation, after setting all properties.
function verespaco_hrextinv_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verespaco_hrextinv (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verespaco_hrextver_Callback(hObject, eventdata, handles)
% hObject    handle to verespaco_hrextver (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_hrextver as text
%        str2double(get(hObject,'String')) returns contents of verespaco_hrextver as
a double

% --- Executes during object creation, after setting all properties.
function verespaco_hrextver_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verespaco_hrextver (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');

```

end

```
function verespaco_caudalar_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_caudalar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verespaco_caudalar as text
%         str2double(get(hObject,'String')) returns contents of verespaco_caudalar as a double
```

```
% --- Executes during object creation, after setting all properties.
function verespaco_caudalar_CreateFcn(hObject, eventdata, handles)
% hObject      handle to verespaco_caudalar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% -----
function verespaco_elementos_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_elementos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function verespaco_espaco_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_espaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function verespaco_horario_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_horario (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function verespaco_verespaco_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_verespaco (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
function verespaco_janela_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_janela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
```

```
function verespaco_porta_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_porta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% -----
```

```
function verespaco_pa_Callback(hObject, eventdata, handles)
% hObject      handle to verespaco_pa (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```

function varargout = ver_parede(varargin)
% VER_PAREDE MATLAB code for ver_parede.fig
%     VER_PAREDE, by itself, creates a new VER_PAREDE or raises the existing
%     singleton*.
%
%     H = VER_PAREDE returns the handle to a new VER_PAREDE or the handle to
%     the existing singleton*.
%
%     VER_PAREDE('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in VER_PAREDE.M with the given input arguments.
%
%     VER_PAREDE('Property','Value',...) creates a new VER_PAREDE or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before ver_parede_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to ver_parede_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help ver_parede

% Last Modified by GUIDE v2.5 21-Jul-2016 11:44:18

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @ver_parede_OpeningFcn, ...
                  'gui_OutputFcn',  @ver_parede_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before ver_parede is made visible.
function ver_parede_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to ver_parede (see VARARGIN)

% Choose default command line output for ver_parede
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```



```

movegui(gcf,'center')

set(handles.figure1,'Name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento')

load projecto.mat

global carproj

if isempty(projecto(carproj).bdados.elementos)==1
    set(handles.verparede_lista,'String','(Não existe nenhum elemento)')
else
    set(handles.verparede_lista,'String',{projecto(carproj).bdados.elementos.nome})
end

% UIWAIT makes ver_parede wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = ver_parede_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in verparede_lista.
function verparede_lista_Callback(hObject, eventdata, handles)
% hObject handle to verparede_lista (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns verparede_lista contents ✓
as cell array
% contents{get(hObject,'Value')} returns selected item from verparede_lista

% --- Executes during object creation, after setting all properties.
function verparede_lista_CreateFcn(hObject, eventdata, handles)
% hObject handle to verparede_lista (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verparede_nome_Callback(hObject, eventdata, handles)
% hObject handle to verparede_nome (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verparede_nome as text
%         str2double(get(hObject,'String')) returns contents of verparede_nome as a double

% --- Executes during object creation, after setting all properties.
function verparede_nome_CreateFcn(hObject, eventdata, handles)
% hObject      handle to verparede_nome (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in verparede_ver.
function verparede_ver_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_ver (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

load projecto.mat

global carproj

if isempty(projecto(carproj).bdados.elementos)==1

    warndlg('Não existem elementos','Aviso !')

else
k = get(handles.verparede_lista,'Value');

set(handles.verparede_nome,'String',projecto(carproj).bdados.elementos(k).nome)
set(handles.verparede_tipo,'String',projecto(carproj).bdados.elementos(k).tipo)
set(handles.verparede_tabela,'Data',projecto(carproj).bdados.elementos(k).tabela)
set(handles.verparede_rsi,'String',projecto(carproj).bdados.elementos(k).rsi)
set(handles.verparede_rse,'String',projecto(carproj).bdados.elementos(k).rse)
set(handles.verparede_u,'String',projecto(carproj).bdados.elementos(k).u)
end

clearvars k

% --- Executes during object creation, after setting all properties.
function verparede_ver_CreateFcn(hObject, eventdata, handles)
% hObject      handle to verparede_ver (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% --- Executes on button press in verparede_eliminar.
function verparede_eliminar_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_eliminar (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
load projecto.mat

global carproj

if isempty(projecto(carproj).bdados.elementos)==1

    warndlg('Não existem elementos','Aviso !')
else

    choice = questdlg('Deseja eliminar o elemento construtivo?', 'Eliminar Elemento', 'Construtivo', 'Sim', 'Não', 'Cancelar', 'Cancelar');

    switch choice

        case 'Sim'

            k = get(handles.verparede_lista, 'Value');

            if k > 23

                projecto(carproj).bdados.elementos(k) = [];

                save projecto.mat projecto
                warndlg('Elemento construtivo eliminado', 'Aviso !')

                set(handles.verparede_lista, 'Value', 1)

                save projecto.mat projecto

                load projecto.mat

                set(handles.verparede_lista, 'String', {projecto(carproj).bdados.elementos.nome})

            else

                errordlg('Não é possível eliminar este elemento construtivo', 'Erro')
            end

        case 'Não'
        case 'Cancelar'
    end

    clearvars choice k

function verparede_rsi_Callback(hObject, eventdata, handles)
% hObject handle to verparede_rsi (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of verparede_rsi as text
% str2double(get(hObject, 'String')) returns contents of verparede_rsi as a
double

```

```

% --- Executes during object creation, after setting all properties.
function verparedede_rsi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verparedede_rsi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verparedede_rse_Callback(hObject, eventdata, handles)
% hObject    handle to verparedede_rse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verparedede_rse as text
%         str2double(get(hObject,'String')) returns contents of verparedede_rse as a
double

% --- Executes during object creation, after setting all properties.
function verparedede_rse_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verparedede_rse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function verparedede_tipo_Callback(hObject, eventdata, handles)
% hObject    handle to verparedede_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verparedede_tipo as text
%         str2double(get(hObject,'String')) returns contents of verparedede_tipo as a
double

% --- Executes during object creation, after setting all properties.
function verparedede_tipo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to verparedede_tipo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

end

```
% -----  
function verparede_inicio_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_inicio (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('inicio.m')  
  
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))  
  
% -----  
function verparede_local_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_local (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('local.m')  
  
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))  
  
% -----  
function verparede_material_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_material (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('registamateriais.m')  
  
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))  
  
% -----  
function verparede_relatorios_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_relatorios (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('relatorios.m')  
  
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))  
  
% -----  
function verparede_adhorario_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_adhorario (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('definir_horario.m')  
  
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))  
  
% -----  
function verparede_defhorario_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_defhorario (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('horario.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))
```

```
% -----  
function verparede_defespaco_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_defespaco (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('definir_espaco.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))
```

```
% -----  
function verparede_verespaco_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_verespaco (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('ver_espaco.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))
```

```
% -----  
function verparede_vidro_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_vidro (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('vidro.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))
```

```
% -----  
function verparede_caixilho_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_caixilho (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('caixilho.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))
```

```
% -----  
function verparede_adparede_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_adparede (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
run('adicionarparede.m')
```

```
close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))
```

```
% -----  
function verparede_adporta_Callback(hObject, eventdata, handles)  
% hObject    handle to verparede_adporta (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB
```

```

% handles      structure with handles and user data (see GUIDATA)
run('Porta.m')

close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))

% -----
function verparede_verporta_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_verporta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('porta_vereliminar.m')

close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))

% -----
function verparede_simples_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_simples (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_simples.m')

close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))

% -----
function verparede_dupla_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_dupla (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_dupla.m')

close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))

% -----
function verparede_duplafolha_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_duplafolha (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_duplafolhamesmoaro.m')

close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))

% -----
function verparede_verjanela_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_verjanela (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
run('janela_vereliminar.m')

close(findobj('type','figure','name','Ver / Eliminar Parede, Teto, Cobertura e Pavimento'))

function verparede_u_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_u (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of verparede_u as text
%        str2double(get(hObject,'String')) returns contents of verparede_u as a
double

% --- Executes during object creation, after setting all properties.
function verparede_u_CreateFcn(hObject, eventdata, handles)
% hObject      handle to verparede_u (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function verparede_elementos_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_elementos (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function verparede_espaco_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_espaco (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function verparede_horario_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_horario (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function verparede_janela_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_janela (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function verparede_porta_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_porta (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function verparede_parede_Callback(hObject, eventdata, handles)
% hObject      handle to verparede_parede (see GCBO)

```



```
% eventdata reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% -----
```

```
function verparede_Callback(hObject, eventdata, handles)
% hObject     handle to verparede (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```

function varargout = vidro(varargin)
% VIDRO MATLAB code for vidro.fig
%     VIDRO, by itself, creates a new VIDRO or raises the existing
%     singleton*.
%
%     H = VIDRO returns the handle to a new VIDRO or the handle to
%     the existing singleton*.
%
%     VIDRO('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in VIDRO.M with the given input arguments.
%
%     VIDRO('Property','Value',...) creates a new VIDRO or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before vidro_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to vidro_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help vidro

% Last Modified by GUIDE v2.5 21-Jul-2016 05:08:20

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @vidro_OpeningFcn, ...
                  'gui_OutputFcn',  @vidro_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before vidro is made visible.
function vidro_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to vidro (see VARARGIN)

% Choose default command line output for vidro
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```

```

movegui(gcf,'center')

set(handles.figure1,'Name','Vidro')

% Para Não aparecer os painéis
set(handles.panel_ver_vidro,'Visible','Off')
set(handles.panel_adicionar_vidro,'Visible','Off')

% Tabela dos vidros sem linhas
set(handles.tabela_vidro,'Data',[])

% Lista de materiais (vidro)

load projecto.mat

global carproj

set(handles.vidro_listamateriais,'String',{'(Selecione)',projecto(carproj).bdados.✓
material.vidros.nome})

% UIWAIT makes vidro wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = vidro_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in vidro_simples.
function vidro_simples_Callback(hObject, eventdata, handles)
% hObject handle to vidro_simples (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
set(handles.vidro_duplo,'Value',0)
set(handles.vidro_triplo,'Value',0)
set(handles.introduzir_Uvidro,'Value',0)
set(handles.vidro_panelsimpduptri,'Visible','On')
set(handles.vidro_paneluvalor,'Visible','Off')

if get(handles.vidro_simples,'Value')== 1
set(handles.vidro_text1,'Visible','On')
set(handles.vidro_text2,'Visible','On')
set(handles.vidro_text3,'Visible','Off')
set(handles.vidro_text13,'Visible','Off')
set(handles.vidro_text14,'Visible','On')
set(handles.vidro_text15,'Visible','On')
set(handles.vidro_text16,'Visible','Off')
set(handles.vidro_text17,'Visible','Off')
set(handles.vidro_rs1,'Visible','Off')

```

```

set(handles.vidro_rs2,'Visible','Off')
set(handles.vidro_rsi,'Visible','On')
set(handles.vidro_rse,'Visible','On')
else
set(handles.vidro_text1,'Visible','Off')
set(handles.vidro_text2,'Visible','Off')
set(handles.vidro_text3,'Visible','Off')
set(handles.vidro_text13,'Visible','Off')
set(handles.vidro_text14,'Visible','Off')
set(handles.vidro_text15,'Visible','Off')
set(handles.vidro_text16,'Visible','Off')
set(handles.vidro_text17,'Visible','Off')
set(handles.vidro_rs1,'Visible','Off')
set(handles.vidro_rs2,'Visible','Off')
set(handles.vidro_rsi,'Visible','Off')
set(handles.vidro_rse,'Visible','Off')
end

% Hint: get(hObject,'Value') returns toggle state of vidro_simples

% --- Executes on button press in vidro_duplo.
function vidro_duplo_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_duplo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.vidro_simples,'Value',0)
set(handles.vidro_triplo,'Value',0)
set(handles.introduzir_Uvidro,'Value',0)
set(handles.vidro_panelsimpduptri,'Visible','On')
set(handles.vidro_paneluvalor,'Visible','Off')

if get(handles.vidro_duplo,'Value')== 1
set(handles.vidro_text1,'Visible','On')
set(handles.vidro_text2,'Visible','On')
set(handles.vidro_text3,'Visible','On')
set(handles.vidro_text13,'Visible','Off')
set(handles.vidro_text14,'Visible','On')
set(handles.vidro_text15,'Visible','On')
set(handles.vidro_text16,'Visible','On')
set(handles.vidro_text17,'Visible','Off')
set(handles.vidro_rs1,'Visible','On')
set(handles.vidro_rs2,'Visible','Off')
set(handles.vidro_rsi,'Visible','On')
set(handles.vidro_rse,'Visible','On')
else
set(handles.vidro_text1,'Visible','Off')
set(handles.vidro_text2,'Visible','Off')
set(handles.vidro_text3,'Visible','Off')
set(handles.vidro_text13,'Visible','Off')
set(handles.vidro_text14,'Visible','Off')
set(handles.vidro_text15,'Visible','Off')
set(handles.vidro_text16,'Visible','Off')
set(handles.vidro_text17,'Visible','Off')
set(handles.vidro_rs1,'Visible','Off')
set(handles.vidro_rs2,'Visible','Off')
set(handles.vidro_rsi,'Visible','Off')
set(handles.vidro_rse,'Visible','Off')
end

```

```
% Hint: get(hObject,'Value') returns toggle state of vidro_duplo
```

```
function vidro_rsi_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_rsi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vidro_rsi as text
%         str2double(get(hObject,'String')) returns contents of vidro_rsi as a double
```

```
% --- Executes during object creation, after setting all properties.
function vidro_rsi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vidro_rsi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function vidro_rse_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_rse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vidro_rse as text
%         str2double(get(hObject,'String')) returns contents of vidro_rse as a double
```

```
% --- Executes during object creation, after setting all properties.
function vidro_rse_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vidro_rse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function vidro_rs1_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_rs1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vidro_rs1 as text
%         str2double(get(hObject,'String')) returns contents of vidro_rs1 as a double
```

```

% --- Executes during object creation, after setting all properties.
function vidro_rsl_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vidro_rsl (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in adicionar_camada_vidro.
function adicionar_camada_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to adicionar_camada_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

load projecto.mat
global carproj

y1 = get(handles.vidro_listamateriais,'Value');

if y1 >1
x1 = projecto(carproj).bdados.material.vidros(y1-1).nome;
x2 = projecto(carproj).bdados.material.vidros(y1-1).k;

old_data = get(handles.tabela_vidro,'Data');
new_row = {x1,x2,''};
new_data = vertcat(old_data,new_row);
set(handles.tabela_vidro,'Data',new_data)
else
errordlg('Escolha o material na lista de vidros','Erro')
end

clearvars y1 x1 old_data new_row new_data

% --- Executes on button press in remover_camada_vidro.
function remover_camada_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to remover_camada_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global tabelavidro

tabelavidro = get(handles.tabela_vidro,'Data');

k = size(tabelavidro,1);

if k > 0

tabelavidro(k,:)=[];
else
end

```

```

set(handles.tabela_vidro, 'Data', tabelavidro)

clearvars k

% --- Executes on button press in guardar_vidro.
function guardar_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to guardar_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

choice = questdlg('Deseja adicionar o vidro?', '', 'Sim', 'Não', 'Cancelar', 'Cancelar');

switch choice

    case 'Sim'
load projecto.mat

global carproj

% Introduzir o valor do coeficiente global de transmissão de calor

if get(handles.introduzir_Uvidro, 'Value')==1

    k = length(projecto(carproj).bdados.vidro)+1;

    projecto(carproj).bdados.vidro(k).nome = get(handles.vidro_nome, 'String');

    switch get(handles.vidro_tipodevidro, 'Value')
        case 1
        case 2
            projecto(carproj).bdados.vidro(k).tipo = 'Vidro simples';
        case 3
            projecto(carproj).bdados.vidro(k).tipo = 'Vidro duplo';
        case 4
            projecto(carproj).bdados.vidro(k).tipo = 'Vidro triplo';
    end

    projecto(carproj).bdados.vidro(k).ug=get(handles.ug_vidro, 'String');

% Vidro Simples
elseif get(handles.vidro_simples, 'Value')== 1

    x1 =str2double(get(handles.vidro_rsi, 'String'));
    x2 =str2double(get(handles.vidro_rse, 'String'));
    x3 =get(handles.tabela_vidro, 'Data');

    ug =ug_simples(x1,x2,x3);

    k = length(projecto(carproj).bdados.vidro)+1;

    projecto(carproj).bdados.vidro(k).nome = get(handles.vidro_nome, 'String');
    projecto(carproj).bdados.vidro(k).tipo = 'Vidro simples';
    projecto(carproj).bdados.vidro(k).ug = ug;

% Vidro duplo
elseif get(handles.vidro_duplo, 'Value')==1

```

```

x1 =str2double(get(handles.vidro_rsi,'String'));
x2 =str2double(get(handles.vidro_rse,'String'));
x3 =get(handles.tabela_vidro,'Data');
x4 =str2double(get(handles.vidro_rsl,'String'));

ug = ug_duplo(x1,x2,x3,x4);

k = length(projecto(carproj).bdados.vidro)+1;

projecto(carproj).bdados.vidro(k).nome = get(handles.vidro_nome,'String');
projecto(carproj).bdados.vidro(k).tipo = 'Vidro duplo';
projecto(carproj).bdados.vidro(k).ug = ug;

%Vidro triplo
elseif get(handles.vidro_triplo,'Value')==1

    x1 =str2double(get(handles.vidro_rsi,'String'));
    x2 =str2double(get(handles.vidro_rse,'String'));
    x3 =get(handles.tabela_vidro,'Data');
    x4 =str2double(get(handles.vidro_rsl,'String'));
    x5 =str2double(get(handles.vidro_rs2,'String'));

    ug = ug_triplo(x1,x2,x3,x4,x5);

    k = length(projecto(carproj).bdados.vidro)+1;

    projecto(carproj).bdados.vidro(k).nome = get(handles.vidro_nome,'String');
    projecto(carproj).bdados.vidro(k).tipo = 'Vidro triplo';
    projecto(carproj).bdados.vidro(k).ug = ug;
end

warndlg('O vidro foi adicionado','Aviso!')

save projecto.mat projecto
    case 'Não'
    case 'Cancelar'
end

clearvars k x1 x2 x3 x4 x5 ug choice

% --- Executes on button press in introduzir_Uvidro.
function introduzir_Uvidro_Callback(hObject, eventdata, handles)
% hObject      handle to introduzir_Uvidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

set(handles.vidro_simples,'Value',0)
set(handles.vidro_duplo,'Value',0)
set(handles.vidro_triplo,'Value',0)
set(handles.vidro_panelsimpduptri,'Visible','Off')
set(handles.vidro_paneluvalor,'Visible','On')

if get(handles.introduzir_Uvidro,'Value') == 1

    set(handles.ug_vidro,'Visible','On')
    set(handles.vidro_text4,'Visible','On')
    set(handles.vidro_text11,'Visible','On')
    set(handles.vidro_text18,'Visible','On')
    set(handles.vidro_tipodevidro,'Visible','On')

```



```

else
    set(handles.ug_vidro, 'Visible', 'Off')
    set(handles.vidro_text4, 'Visible', 'Off')
    set(handles.vidro_text11, 'Visible', 'Off')
    set(handles.vidro_text18, 'Visible', 'Off')
    set(handles.vidro_tipodevidro, 'Visible', 'Off')
end
% Hint: get(hObject, 'Value') returns toggle state of introduzir_Uvidro

% --- Executes on button press in adividro_panel.
function adividro_panel_Callback(hObject, eventdata, handles)
% hObject    handle to adividro_panel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.figure1, 'Name', 'Adicionar Vidro')

set(handles.panel_ver_vidro, 'Visible', 'Off')
set(handles.panel_adicionar_vidro, 'Visible', 'On')
set(handles.vidro_text6, 'Visible', 'On')
set(handles.vidro_nome, 'Visible', 'On')

% --- Executes on button press in verelim_vidro.
function verelim_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to verelim_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.figure1, 'Name', 'Ver / Eliminar Vidro')

set(handles.vidro_text6, 'Visible', 'Off')
set(handles.vidro_nome, 'Visible', 'Off')
set(handles.panel_ver_vidro, 'Visible', 'On')
set(handles.panel_adicionar_vidro, 'Visible', 'Off')

load projecto.mat

global carproj
set(handles.lista_vidros, 'String', {projecto(carproj).bdados.vidro.nome})

function vidro_nome_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_nome (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of vidro_nome as text
%        str2double(get(hObject, 'String')) returns contents of vidro_nome as a
double

% --- Executes during object creation, after setting all properties.
function vidro_nome_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vidro_nome (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

```

```

%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in lista_vidros.
function lista_vidros_Callback(hObject, eventdata, handles)
% hObject      handle to lista_vidros (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns lista_vidros contents as a
cell array
%         contents{get(hObject,'Value')} returns selected item from lista_vidros

% --- Executes during object creation, after setting all properties.
function lista_vidros_CreateFcn(hObject, eventdata, handles)
% hObject      handle to lista_vidros (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function vidronome_Callback(hObject, eventdata, handles)
% hObject      handle to vidronome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vidronome as text
%         str2double(get(hObject,'String')) returns contents of vidronome as a double

% --- Executes during object creation, after setting all properties.
function vidronome_CreateFcn(hObject, eventdata, handles)
% hObject      handle to vidronome (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in ver_vidro.
function ver_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to ver_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

load projecto.mat
global carproj

k = get(handles.lista_vidros, 'Value');

set(handles.vidronome, 'String', projecto(carproj).bdados.vidro(k).nome)
set(handles.tipo_vidro, 'String', projecto(carproj).bdados.vidro(k).tipo)
set(handles.coef_vidro, 'String', projecto(carproj).bdados.vidro(k).ug)

clearvars k

% --- Executes on button press in eliminar_vidro.
function eliminar_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to eliminar_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

load projecto.mat
global carproj

if get(handles.lista_vidros, 'Value') > 5

    choice = questdlg('Tem a certeza que deseja eliminar o ✓  
vidro', 'Atenção!', 'Sim', 'Não', 'Cancelar', 'Cancelar');

    switch choice
        case 'Sim'

            k = get(handles.lista_vidros, 'Value');

            projecto(carproj).bdados.vidro(k) = [];

            save projecto.mat projecto

            set(handles.lista_vidros, 'String', {projecto(carproj).bdados.vidro.nome})
            set(handles.lista_vidros, 'Value', 1)
            warndlg('Vidro eliminado', 'Atenção!')

            set(handles.vidronome, 'String', '')
            set(handles.tipo_vidro, 'String', '')
            set(handles.coef_vidro, 'String', '')
            case 'Não'
            case 'Cancelar'
        end
    else
        errordlg('Não pode eliminar este vidro', 'Aviso')
    end

clearvars k choice

function tipo_vidro_Callback(hObject, eventdata, handles)
% hObject      handle to tipo_vidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of tipo_vidro as text

```

```

%         str2double(get(hObject,'String')) returns contents of tipo_vidro as a
double

% --- Executes during object creation, after setting all properties.
function tipo_vidro_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tipo_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function coef_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to coef_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of coef_vidro as text
%         str2double(get(hObject,'String')) returns contents of coef_vidro as a
double

% --- Executes during object creation, after setting all properties.
function coef_vidro_CreateFcn(hObject, eventdata, handles)
% hObject    handle to coef_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function vidro_inicio_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_inicio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('inicio.m')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_local_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_local (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

run('local.m')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_materiais_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_materiais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('registamateriais.m')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_elementos_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_elementos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function vidro_horarios_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_horarios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function vidro_relatorios_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_relatorios (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('relatorios.m')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_vidro_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_vidro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function vidro_caixilho_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_caixilho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('caixilho.m')

```

```

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_adiporta_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_adiporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('Porta')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_verporta_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_verporta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('porta_vereliminair')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_jansimples_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_jansimples (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('janela_simples')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_jandupla_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_jandupla (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('janela_dupla')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_duplafolha_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_duplafolha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

run('janela_duplafolhamesmoaro')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_verjanela_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_verjanela (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('janela_vereliminar')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_defespaco_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_defespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('definir_espaco')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_verespaco_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_verespaco (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

run('ver_espaco')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_adparede_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_adparede (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
run('adicionarparede')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_verparede_Callback(hObject, eventdata, handles)
% hObject    handle to vidro_verparede (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

run('ver_parede')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_adhorario_Callback(hObject, eventdata, handles)
% hObject handle to vidro_adhorario (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

run('definir_horario')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

% -----
function vidro_defhorario_Callback(hObject, eventdata, handles)
% hObject handle to vidro_defhorario (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

run('horario')

set(handles.figure1,'Name','Vidro')

close(findobj('type','figure','name','Vidro'))

function ug_vidro_Callback(hObject, eventdata, handles)
% hObject handle to ug_vidro (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ug_vidro as text
% str2double(get(hObject,'String')) returns contents of ug_vidro as a double

% --- Executes during object creation, after setting all properties.
function ug_vidro_CreateFcn(hObject, eventdata, handles)
% hObject handle to ug_vidro (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

```



```

% --- Executes on selection change in vidro_listamateriais.
function vidro_listamateriais_Callback(hObject, eventdata, handles)
% hObject      handle to vidro_listamateriais (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns vidro_listamateriais✓
%           contents as cell array
%           contents{get(hObject,'Value')} returns selected item from✓
vidro_listamateriais

% --- Executes during object creation, after setting all properties.
function vidro_listamateriais_CreateFcn(hObject, eventdata, handles)
% hObject      handle to vidro_listamateriais (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in vidro_tipodevidro.
function vidro_tipodevidro_Callback(hObject, eventdata, handles)
% hObject      handle to vidro_tipodevidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns vidro_tipodevidro✓
%           contents as cell array
%           contents{get(hObject,'Value')} returns selected item from vidro_tipodevidro

% --- Executes during object creation, after setting all properties.
function vidro_tipodevidro_CreateFcn(hObject, eventdata, handles)
% hObject      handle to vidro_tipodevidro (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get✓
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in vidro_triplo.
function vidro_triplo_Callback(hObject, eventdata, handles)
% hObject      handle to vidro_triplo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of vidro_triplo

set(handles.vidro_simples,'Value',0)

```

```

set(handles.vidro_duplo,'Value',0)
set(handles.introduzir_Uvidro,'Value',0)
set(handles.vidro_panelsimpduptri,'Visible','On')
set(handles.vidro_paneluvalor,'Visible','Off')

if get(handles.vidro_triplo,'Value')== 1
set(handles.vidro_text1,'Visible','On')
set(handles.vidro_text2,'Visible','On')
set(handles.vidro_text3,'Visible','On')
set(handles.vidro_text13,'Visible','On')
set(handles.vidro_text14,'Visible','On')
set(handles.vidro_text15,'Visible','On')
set(handles.vidro_text16,'Visible','On')
set(handles.vidro_text17,'Visible','On')
set(handles.vidro_rs1,'Visible','On')
set(handles.vidro_rs2,'Visible','On')
set(handles.vidro_rsi,'Visible','On')
set(handles.vidro_rse,'Visible','On')
else
set(handles.vidro_text1,'Visible','Off')
set(handles.vidro_text2,'Visible','Off')
set(handles.vidro_text3,'Visible','Off')
set(handles.vidro_text13,'Visible','Off')
set(handles.vidro_text14,'Visible','Off')
set(handles.vidro_text15,'Visible','Off')
set(handles.vidro_text16,'Visible','Off')
set(handles.vidro_text17,'Visible','Off')
set(handles.vidro_rs1,'Visible','Off')
set(handles.vidro_rs2,'Visible','Off')
set(handles.vidro_rsi,'Visible','Off')
set(handles.vidro_rse,'Visible','Off')
end

function vidro_rs2_Callback(hObject, eventdata, handles)
% hObject      handle to vidro_rs2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vidro_rs2 as text
%          str2double(get(hObject,'String')) returns contents of vidro_rs2 as a double

% --- Executes during object creation, after setting all properties.
function vidro_rs2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to vidro_rs2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function vidro_espaco_Callback(hObject, eventdata, handles)
% hObject      handle to vidro_espaco (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% -----
function vidro_janela_Callback(hObject, eventdata, handles)
% hObject     handle to vidro_janela (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% -----
function vidro_porta_Callback(hObject, eventdata, handles)
% hObject     handle to vidro_porta (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% -----
function vidro_parede_Callback(hObject, eventdata, handles)
% hObject     handle to vidro_parede (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```



## **Anexo C – Funções**



```

% Função que determina pressão atmosférica com o efeito de altitude

%pb - Pressão estática (ao nível do mar)
%Tb - Temperatura normalizada (ao nível do mar)
%hb - Altitude do local

function[resultado]=altitude_pressao(pb,Tb,hb)

R = 8.31432;
g0 = 9.80665;
M = 0.0289644;
Lb=-0.0065;

x = (-R*Lb) / (g0*M);

y = (-g0*M) / (R*Lb);

syms h p

E1 = hb-((Tb+273.15)/Lb)*(((p/pb)^x)-1);

E2 = pb*(1+((Lb/(Tb+273.15))*(h-hb)))^y;

m = solve([E1,E2],[h,p]);
resultado = vpa(m.p);

end

```

```
% Função que determina a primeira linha do dia em análise

function[z]=analise_dia(mesb,diab)

x=mesb-1;
y=diab;

dia = 1;      % dia inicial
mes = 1;      % mes inicial
ano = 2015;   % ano inicial
di = datenum([ano mes dia 0 0 0]);    % dia inicial em inteiro
df = datenum([ano+1 mes dia 0 0 0])-1; % dia final em inteiro

dint=datenum([ano mes+x dia+y 0 0 0]);

z= (dint-di)*24;

end
```



```

% Função que faz o cálculo das cargas térmicas ao longo do ano todo

% diasemana - Dia da semana no primeiro dia do ano

function [tabela] = analise_hora(diasemana)

load projecto.mat

global carproj

for i=1:length(projecto(carproj).espaco);

% Configura as 2 primeiras colunas
load dia_hora2.mat
dia = dia_hora(:,1);
hora = dia_hora(:,2);

%Define os dias da semana/feridos
dia_semana=di_semana(diasemana);

% Informação sobre os horario utilizados no espaço
[horario_ocupacao,horario_iluminacao,horario_equipamentos,horario_diversas]✓
=def_horarios(i);

% Coeficientes obtidos do horários
[coef_ocupacao,coef_iluminacao,coef_equipamentos,coef_diversas]=coef_horarios✓
(horario_ocupacao,horario_iluminacao,horario_equipamentos,horario_diversas,✓
dia_semana);

% Obter a pressão do local através da altitude
pb = 101325; % Pa
Tb = 15; %°C
hb = str2double(projecto(carproj).local.altitude);

p=altitude_pressao(pb,Tb,hb);

T=str2double(projecto(carproj).espaco(i).temp_interior);
HR=str2double(projecto(carproj).espaco(i).hum_interior);
entalpia_int = calc_entalpia(T,HR,p);

% Alocar memoria:

carga_ocupacao_sensivel(1:8760,1)=zeros;
carga_ocupacao_latente(1:8760,1)=zeros;
carga_iluminacao(1:8760,1)=zeros;
carga_eletricos(1:8760,1)=zeros;
carga_diversa_sensivel(1:8760,1)=zeros;
carga_diversa_latente(1:8760,1)=zeros;
carga_infiltracoes_sensivel(1:8760,1)=zeros;
carga_infiltracoes_total(1:8760,1)=zeros;
carga_infiltracoes_latente(1:8760,1)=zeros;
carga_sensivel_arrefe(1:8760)=zeros;
carga_latente_arrefe(1:8760)=zeros;
carga_total_arrefe(1:8760)=zeros;
carga_sensivel_aque(1:8760)=zeros;
carga_latente_aque(1:8760)=zeros;
carga_total_aque(1:8760)=zeros;

```

```

tic

for k=1:8760

    disp(k)
    % Ocupação
    carga_ocupacao_sensivel(k,1) = projecto(carproj).espaco(i).ocupacao.quant *✓
    str2double(projecto(carproj).espaco(i).ocupacao.csensivel)...
        *(coef_ocupacao(k,1)/100);
    carga_ocupacao_latente(k,1) = projecto(carproj).espaco(i).ocupacao.quant *✓
    str2double(projecto(carproj).espaco(i).ocupacao.clatente)...
        *(coef_ocupacao(k,1)/100);

    % Iluminação
    carga_iluminacao(k,1) = projecto(carproj).espaco(i).ilumina*(coef_iluminacao(k,✓
    1)/100);

    % Equipamentos Elétricos
    carga_eletricos(k,1) = projecto(carproj).espaco(i).eletricos*(coef_equipamentos✓
    (k,1)/100);

    % Diversas
    carga_diversa_sensivel(k,1) = str2double(projecto(carproj).espaco(i).diversas.✓
    sensivel)*(coef_diversas(k,1)/100);
    carga_diversa_latente(k,1) = str2double(projecto(carproj).espaco(i).diversas.✓
    latente)*(coef_diversas(k,1)/100);

    % Carga devida a Infiltrações

    switch length(projecto(carproj).local.ficheiro(1,:))

        case 6 % .dat

            T= projecto(carproj).local.ficheiro(k,3);
            if k == 8760
                T = 0;
            else
                T= projecto(carproj).local.ficheiro(k,3);
            end
            HR=projecto(carproj).local.ficheiro(k,4);
            entalpia(k,1) = calc_entalpia(T,HR,p);
            carga_infiltracoes_sensivel(k,1) = 1.23*str2double(projecto(carproj).espaco(i).✓
            infiltracoes.caudal)*(projecto(carproj).local.ficheiro(k,3)-str2double(projecto✓
            (carproj).espaco(i).temp_interior));
            carga_infiltracoes_total(k,1) = 1.20*str2double(projecto(carproj).espaco(i).✓
            infiltracoes.caudal)*(entalpia(k,1)-entalpia_int);
            carga_infiltracoes_latente(k,1) = carga_infiltracoes_total(k,1)-✓
            carga_infiltracoes_sensivel(k,1);
            case 7 % .epw

                T= projecto(carproj).local.ficheiro(k,4);
                HR=projecto(carproj).local.ficheiro(k,5);
                entalpia(k,1) = calc_entalpia(T,HR,p);
                carga_infiltracoes_sensivel(k,1) = 1.23*str2double(projecto(carproj).espaco(i).✓
                infiltracoes.caudal)*(projecto(carproj).local.ficheiro(k,4)-str2double(projecto✓
                (carproj).espaco(i).temp_interior));
                carga_infiltracoes_total(k,1) = 1.20*str2double(projecto(carproj).espaco(i).✓
                infiltracoes.caudal)*(entalpia(k,1)-entalpia_int);
                carga_infiltracoes_latente(k,1) = carga_infiltracoes_total(k,1)-✓

```

```

carga_infiltracoes_sensivel(k,1);
    end

    %Ganhos Externos

    for y = 1:length(projecto(carproj).espaco(i).ganhos_ext)

        switch length(projecto(carproj).local.ficheiro(1,:))

            case 6 % .dat

                ganhos_ext(k,y)={str2double(projecto(carproj).espaco(i).ganhos_ext(y).✓
quant)*str2double(projecto(carproj).espaco(i).ganhos_ext(y).area)...
                *projecto(carproj).espaco(i).ganhos_ext(y).u*(projecto(carproj).✓
local.ficheiro(k,3) - str2double(projecto(carproj).espaco(i).temp_interior))});

            case 7 % .epw

                ganhos_ext(k,y)={str2double(projecto(carproj).espaco(i).ganhos_ext(y).✓
quant)*str2double(projecto(carproj).espaco(i).ganhos_ext(y).area)...
                *projecto(carproj).espaco(i).ganhos_ext(y).u*(projecto(carproj).✓
local.ficheiro(k,4)- str2double(projecto(carproj).espaco(i).temp_interior))});
            end
        end

        %Ganhos internos
        %Elementos construtivos

        for z = 1:length(projecto(carproj).espaco(i).ganhos_int)

            switch length(projecto(carproj).local.ficheiro(1,:))

                case 6 % .dat

                    ganhos_int(k,z) = {str2double(projecto(carproj).espaco(i).ganhos_int(z).✓
quant)*str2double(projecto(carproj).espaco(i).ganhos_int(z).area)...
                    *projecto(carproj).espaco(i).ganhos_int(z).u*(str2double(projecto✓
(carproj).espaco(i).ganhos_int(z).temp) - str2double(projecto(carproj).espaco(i).✓
temp_interior))});

                case 7 % .epw

                    ganhos_int(k,z) = {str2double(projecto(carproj).espaco(i).ganhos_int(z).✓
quant)*str2double(projecto(carproj).espaco(i).ganhos_int(z).area)...
                    *projecto(carproj).espaco(i).ganhos_int(z).u*(str2double(projecto✓
(carproj).espaco(i).ganhos_int(z).temp)- str2double(projecto(carproj).espaco(i).✓
temp_interior))});
                end
            end

            % Cargas Térmicas de Arrefecimento

            carga_sensivel_arrefe(k)=carga_ocupacao_sensivel(k,1)+carga_iluminacao(k,1)✓
+carga_eletricos(k,1)+carga_diversa_sensivel(k,1)...
            +carga_infiltracoes_sensivel(k,1)+sum(cell2mat(ganhos_ext(k,1:end)))+sum✓
(cell2mat(ganhos_int(k,1:end)));
            carga_latente_arrefe(k) =carga_ocupacao_latente(k,1)+carga_diversa_latente(k,1)✓
+carga_infiltracoes_latente(k,1);
            carga_total_arrefe(k) =carga_sensivel_arrefe(k)+ carga_latente_arrefe(k);

            % Cargas térmicas de aquecimento

```

```

for n=1:length(ganhos_int(1,:))

    if ganhos_int{k,n}>0

        ganhos_intinv{k,n}=0;

    else
        ganhos_intinv{k,n}=ganhos_int{k,n};
    end
end

carga_sensivel_aque(k)=carga_infiltracoes_sensivel(k,1)+sum(cell2mat(ganhos_ext✓
(k,1:end)))+sum(cell2mat(ganhos_intinv(k,1:end))));
carga_latente_aque(k) =carga_infiltracoes_latente(k,1);
carga_total_aque(k)    =carga_sensivel_aque(k)+ carga_latente_aque(k);

end

tabela = table(dia(1:k),hora(1:k),dia_semana(1:k),coef_ocupacao(1:k),coef_iluminacao✓
(1:k),coef_equipamentos(1:k),coef_diversas(1:k),carga_ocupacao_sensivel(1:k),✓
carga_ocupacao_latente(1:k),carga_iluminacao,carga_eletricos,...
    carga_diversa_sensivel,carga_diversa_latente,carga_infiltracoes_sensivel,double✓
(carga_infiltracoes_total),double(carga_infiltracoes_latente),ganhos_ext,✓
ganhos_int,...
    (carga_sensivel_arrefe)',double((carga_latente_arrefe)'),double✓
((carga_total_arrefe)'),(carga_sensivel_aque)',double((carga_latente_aque)'),double✓
((carga_total_aque)')));

save tabela.mat tabela

projecto(carproj).relatorios(i)= load('tabela.mat');

save projecto.mat projecto

delete tabela.mat

end

toc

end

```

```
% Função que calcula entalpias

%T - temperatura
%HR - humidade relativa
%p - pressao em Pa

function[h]=calc_entalpia(T,HR,p)

addpath('C:\Users\João Gonçalves\Documents\Novo Projecto\Aplicação\base_dados_originais')
load tabela_pvapor.mat

ps = (myinterp1(T))*10^5;

pv=(HR/100)*ps;

pa=p-pv;

w = 0.622*(pv/pa);

ha = 1.006*(T-0);

hv = 2500.7+(1.84*T);

h = ha+(w*hv);
end
```

```
% Função que vai buscar os respectivos coeficientes dos horarios
```

```
function[coef_ocupacao,coef_iluminacao,coef_equipamentos,coef_diversas]✓  
=coef_horarios(horario_ocupacao,horario_iluminacao,horario_equipamentos,✓  
horario_diversas,dia_semana)
```

```
load projecto.mat
```

```
global carproj
```

```
% Ocupação
```

```
for k = 1:24:8760
```

```
    choice = dia_semana{k,1};
```

```
    x = horario_ocupacao(k,1);
```

```
    switch choice
```

```
        case 'Segunda'
```

```
            [coef_ocupacao(k:k+23,1)] = projecto(carproj).bdados.horario.ocupacao(x).✓  
data(1:24,1);
```

```
        case 'Terça'
```

```
            [coef_ocupacao(k:k+23,1)] = projecto(carproj).bdados.horario.ocupacao(x).✓  
data(1:24,2);
```

```
        case 'Quarta'
```

```
            [coef_ocupacao(k:k+23,1)] = projecto(carproj).bdados.horario.ocupacao(x).✓  
data(1:24,3);
```

```
        case 'Quinta'
```

```
            [coef_ocupacao(k:k+23,1)] = projecto(carproj).bdados.horario.ocupacao(x).✓  
data(1:24,4);
```

```
        case 'Sexta'
```

```
            [coef_ocupacao(k:k+23,1)] = projecto(carproj).bdados.horario.ocupacao(x).✓  
data(1:24,5);
```

```
        case 'Sábado'
```

```
            [coef_ocupacao(k:k+23,1)] = projecto(carproj).bdados.horario.ocupacao(x).✓  
data(1:24,6);
```

```
        case 'Domingo'
```

```
            [coef_ocupacao(k:k+23,1)] = projecto(carproj).bdados.horario.ocupacao(x).✓  
data(1:24,7);
```

```
        case 'Feriado'
```

```
            [coef_ocupacao(k:k+23,1)] = projecto(carproj).bdados.horario.ocupacao(x).✓  
data(1:24,8);
```

```
    end
```

```
end
```

```
% Iluminação
```

```
for k = 1:24:8760
```

```
    choice = dia_semana{k,1};
```

```
    x = horario_iluminacao(k,1);
```

```
    switch choice
```

```
        case 'Segunda'
```

```
            [coef_iluminacao(k:k+23,1)] = projecto(carproj).bdados.horario.iluminacao✓  
(x).data(1:24,1);
```

```

        case 'Terça'
            [coef_iluminacao(k:k+23,1)] = projecto(carproj).bdados.horario.iluminacao✓
(x).data(1:24,2);
        case 'Quarta'
            [coef_iluminacao(k:k+23,1)] = projecto(carproj).bdados.horario.iluminacao✓
(x).data(1:24,3);
        case 'Quinta'
            [coef_iluminacao(k:k+23,1)] = projecto(carproj).bdados.horario.iluminacao✓
(x).data(1:24,4);
        case 'Sexta'
            [coef_iluminacao(k:k+23,1)] = projecto(carproj).bdados.horario.iluminacao✓
(x).data(1:24,5);
        case 'Sábado'
            [coef_iluminacao(k:k+23,1)] = projecto(carproj).bdados.horario.iluminacao✓
(x).data(1:24,6);
        case 'Domingo'
            [coef_iluminacao(k:k+23,1)] = projecto(carproj).bdados.horario.iluminacao✓
(x).data(1:24,7);
        case 'Feriado'
            [coef_iluminacao(k:k+23,1)] = projecto(carproj).bdados.horario.iluminacao✓
(x).data(1:24,8);
    end
end

```

% Equipamentos

```
for k = 1:24:8760
```

```
    choice = dia_semana{k,1};
```

```
    x = horario_equipamentos(k,1);
```

```

    switch choice
        case 'Segunda'
            [coef_equipamentos(k:k+23,1)] = projecto(carproj).bdados.horario.✓
equipamentos(x).data(1:24,1);
        case 'Terça'
            [coef_equipamentos(k:k+23,1)] = projecto(carproj).bdados.horario.✓
equipamentos(x).data(1:24,2);
        case 'Quarta'
            [coef_equipamentos(k:k+23,1)] = projecto(carproj).bdados.horario.✓
equipamentos(x).data(1:24,3);
        case 'Quinta'
            [coef_equipamentos(k:k+23,1)] = projecto(carproj).bdados.horario.✓
equipamentos(x).data(1:24,4);
        case 'Sexta'
            [coef_equipamentos(k:k+23,1)] = projecto(carproj).bdados.horario.✓
equipamentos(x).data(1:24,5);
        case 'Sábado'
            [coef_equipamentos(k:k+23,1)] = projecto(carproj).bdados.horario.✓
equipamentos(x).data(1:24,6);
        case 'Domingo'
            [coef_equipamentos(k:k+23,1)] = projecto(carproj).bdados.horario.✓
equipamentos(x).data(1:24,7);
        case 'Feriado'
            [coef_equipamentos(k:k+23,1)] = projecto(carproj).bdados.horario.✓
equipamentos(x).data(1:24,8);
    end
end

```

```

% Diversas

for k = 1:24:8760

    choice = dia_semana{k,1};

    x = horario_diversas(k,1);

    switch choice
        case 'Segunda'
            [coef_diversas(k:k+23,1)] = proyecto(carproj).bdados.horario.diversas(x).✓
data(1:24,1);
        case 'Terça'
            [coef_diversas(k:k+23,1)] = proyecto(carproj).bdados.horario.diversas(x).✓
data(1:24,2);
        case 'Quarta'
            [coef_diversas(k:k+23,1)] = proyecto(carproj).bdados.horario.diversas(x).✓
data(1:24,3);
        case 'Quinta'
            [coef_diversas(k:k+23,1)] = proyecto(carproj).bdados.horario.diversas(x).✓
data(1:24,4);
        case 'Sexta'
            [coef_diversas(k:k+23,1)] = proyecto(carproj).bdados.horario.diversas(x).✓
data(1:24,5);
        case 'Sábado'
            [coef_diversas(k:k+23,1)] = proyecto(carproj).bdados.horario.diversas(x).✓
data(1:24,6);
        case 'Domingo'
            [coef_diversas(k:k+23,1)] = proyecto(carproj).bdados.horario.diversas(x).✓
data(1:24,7);
        case 'Feriado'
            [coef_diversas(k:k+23,1)] = proyecto(carproj).bdados.horario.diversas(x).✓
data(1:24,8);
    end
end
end

```



```

% Função que interliga os 4 horarios de cada espaço por cada mes.

% i - Número do espaço

function[horario_ocupacao,horario_iluminacao,horario_equipamentos,horario_diversas]✓
=def_horarios(i)

% Definição de Horarios

load projecto.mat

global carproj

%Ocupação

ocupacao(1:744,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela(1,2);%✓
Janeiro
ocupacao(745:1416,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela(1,3);%✓
Fevereiro
ocupacao(1417:2160,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela(1,4);%✓
Março
ocupacao(2161:2880,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela(1,5);%✓
Abril
ocupacao(2881:3624,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela(1,6);%✓
Maio
ocupacao(3625:4344,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela(1,7);%✓
Junho
ocupacao(4345:5088,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela(1,8);%✓
Julho
ocupacao(5089:5832,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela(1,9);%✓
Agosto
ocupacao(5833:6552,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela✓
(1,10);%Setembro
ocupacao(6553:7296,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela✓
(1,11);%Outubro
ocupacao(7297:8016,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela✓
(1,12);%Novembro
ocupacao(8017:8760,1)=projecto(carproj).espaco(i).horario.ocupacao.tabela✓
(1,13);%Dezembro

for k = 1:8760

    if      str2double(ocupacao(k,1))==1

        horario_ocupacao(k,1)=projecto(carproj).espaco(i).horario.ocupacao.um;

    elseif  str2double(ocupacao(k,1))==2

        horario_ocupacao(k,1)=projecto(carproj).espaco(i).horario.ocupacao.dois;

    elseif  str2double(ocupacao(k,1))==3

        horario_ocupacao(k,1)=projecto(carproj).espaco(i).horario.ocupacao.tres;

    elseif  str2double(ocupacao(k,1))==4

        horario_ocupacao(k,1)=projecto(carproj).espaco(i).horario.ocupacao.✓
quatro;

```

```

        end
    end

    % Iluminacao

    iluminacao(1:744,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela(1,2);%✓
Janeiro
    iluminacao(745:1416,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,3);%Fevereiro
    iluminacao(1417:2160,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,4);%Março
    iluminacao(2161:2880,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,5);%Abril
    iluminacao(2881:3624,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,6);%Maio
    iluminacao(3625:4344,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,7);%Junho
    iluminacao(4345:5088,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,8);%Julho
    iluminacao(5089:5832,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,9);%Agosto
    iluminacao(5833:6552,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,10);%Setembro
    iluminacao(6553:7296,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,11);%Outubro
    iluminacao(7297:8016,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,12);%Novembro
    iluminacao(8017:8760,1)=projecto(carproj).espaco(i).horario.iluminacao.tabela✓
(1,13);%Dezembro

    for k = 1:8760

        if      str2double(iluminacao(k,1))==1

            horario_iluminacao(k,1)=projecto(carproj).espaco(i).horario.iluminacao.✓
um;

        elseif str2double(iluminacao(k,1))==2

            horario_iluminacao(k,1)=projecto(carproj).espaco(i).horario.iluminacao.✓
dois;

        elseif str2double(iluminacao(k,1))==3

            horario_iluminacao(k,1)=projecto(carproj).espaco(i).horario.iluminacao.✓
tres;

        elseif str2double(iluminacao(k,1))==4

            horario_iluminacao(k,1)=projecto(carproj).espaco(i).horario.iluminacao.✓
quatro;

        end
    end

    % Equipmentos

```

```

    equipamentos(1:744,1)=projecto(carproj).espaco(i).horario.equipamentos.tabela✓
(1,2);%Janeiro
    equipamentos(745:1416,1)=projecto(carproj).espaco(i).horario.equipamentos.tabela✓
(1,3);%Fevereiro
    equipamentos(1417:2160,1)=projecto(carproj).espaco(i).horario.equipamentos.✓
tabela(1,4);%Março
    equipamentos(2161:2880,1)=projecto(carproj).espaco(i).horario.equipamentos.✓
tabela(1,5);%Abril
    equipamentos(2881:3624,1)=projecto(carproj).espaco(i).horario.equipamentos.✓
tabela(1,6);%Maio
    equipamentos(3625:4344,1)=projecto(carproj).espaco(i).horario.equipamentos.✓
tabela(1,7);%Junho
    equipamentos(4345:5088,1)=projecto(carproj).espaco(i).horario.equipamentos.✓
tabela(1,8);%Julho
    equipamentos(5089:5832,1)=projecto(carproj).espaco(i).horario.equipamentos.✓
tabela(1,9);%Agosto
    equipamentos(5833:6552,1)=projecto(carproj).espaco(i).horario.equipamentos.✓
tabela(1,10);%Setembro
    equipamentos(6553:7296,1)=projecto(carproj).espaco(i).horario.equipamentos.✓
tabela(1,11);%Outubro
    equipamentos(7297:8016,1)=projecto(carproj).espaco(i).horario.equipamentos.✓
tabela(1,12);%Novembro
    equipamentos(8017:8760,1)=projecto(carproj).espaco(i).horario.equipamentos.✓
tabela(1,13);%Dezembro

    for k = 1:8760

        if      str2double(equipamentos(k,1))==1

            horario_equipamentos(k,1)=projecto(carproj).espaco(i).horario.✓
equipamentos.um;

            elseif str2double(equipamentos(k,1))==2

                horario_equipamentos(k,1)=projecto(carproj).espaco(i).horario.✓
equipamentos.dois;

            elseif str2double(equipamentos(k,1))==3

                horario_equipamentos(k,1)=projecto(carproj).espaco(i).horario.✓
equipamentos.tres;

            elseif str2double(equipamentos(k,1))==4

                horario_equipamentos(k,1)=projecto(carproj).espaco(i).horario.✓
equipamentos.quatro;

        end
    end

    % Diversas

    diversas(1:744,1)=projecto(carproj).espaco(i).horario.diversas.tabela(1,2);%✓
Janeiro
    diversas(745:1416,1)=projecto(carproj).espaco(i).horario.diversas.tabela(1,3);%✓
Fevereiro
    diversas(1417:2160,1)=projecto(carproj).espaco(i).horario.diversas.tabela(1,4);%✓
Março
    diversas(2161:2880,1)=projecto(carproj).espaco(i).horario.diversas.tabela(1,5);%✓
Abril

```

```

diversas(2881:3624,1)=projecto(carproj).espaco(i).horario.diversas.tabela(1,6);%✓
Maio
diversas(3625:4344,1)=projecto(carproj).espaco(i).horario.diversas.tabela(1,7);%✓
Junho
diversas(4345:5088,1)=projecto(carproj).espaco(i).horario.diversas.tabela(1,8);%✓
Julho
diversas(5089:5832,1)=projecto(carproj).espaco(i).horario.diversas.tabela(1,9);%✓
Agosto
diversas(5833:6552,1)=projecto(carproj).espaco(i).horario.diversas.tabela✓
(1,10);%Setembro
diversas(6553:7296,1)=projecto(carproj).espaco(i).horario.diversas.tabela✓
(1,11);%Outubro
diversas(7297:8016,1)=projecto(carproj).espaco(i).horario.diversas.tabela✓
(1,12);%Novembro
diversas(8017:8760,1)=projecto(carproj).espaco(i).horario.diversas.tabela✓
(1,13);%Dezembro

for k = 1:8760

    if      str2double(diversas(k,1))==1

        horario_diversas(k,1)=projecto(carproj).espaco(i).horario.diversas.um;

    elseif str2double(diversas(k,1))==2

        horario_diversas(k,1)=projecto(carproj).espaco(i).horario.diversas.dois;

    elseif str2double(diversas(k,1))==3

        horario_diversas(k,1)=projecto(carproj).espaco(i).horario.diversas.tres;

    elseif str2double(diversas(k,1))==4

        horario_diversas(k,1)=projecto(carproj).espaco(i).horario.diversas.✓
quatro;

    end
end

```

```
% Função que define qual é o dia da semana ao longo do ano. E define os  
% feriados
```

```
% diasemana - Dia da semana do primeiro dia do ano
```

```
function[ano_fim]=di_semana(diasemana)
```

```
segunda ={'Segunda'};  
segunda = repmat(segunda,24,1);  
terca ={'Terça'};  
terca = repmat(terca,24,1);  
quarta ={'Quarta'};  
quarta = repmat(quarta,24,1);  
quinta ={'Quinta'};  
quinta = repmat(quinta,24,1);  
sexta ={'Sexta'};  
sexta = repmat(sexta,24,1);  
sabado ={'Sábado'};  
sabado = repmat(sabado,24,1);  
domingo ={'Domingo'};  
domingo = repmat(domingo,24,1);
```

```
switch diasemana
```

```
case 1 %Segunda
```

```
    semana = [segunda;terca;quarta;quinta;sexta;sabado;domingo];
```

```
    ano=repmat(semana,52,1);
```

```
    ano_fim= [ano;segunda];
```

```
case 2 %Terça
```

```
    semana = [terca;quarta;quinta;sexta;sabado;domingo;segunda];
```

```
    ano=repmat(semana,52,1);
```

```
    ano_fim= [ano;terca];
```

```
case 3 %Quarta
```

```
    semana = [quarta;quinta;sexta;sabado;domingo;segunda;terca];
```

```
    ano=repmat(semana,52,1);
```

```
    ano_fim= [ano;quarta];
```

```
case 4 %Quinta
```

```
    semana = [quinta;sexta;sabado;domingo;segunda;terca;quarta];
```

```
    ano=repmat(semana,52,1);
```

```
    ano_fim= [ano;quinta];
```

```
case 5 %Sexta
```

```
    semana = [sexta;sabado;domingo;segunda;terca;quarta;quinta];
```

```
    ano=repmat(semana,52,1);
```

```
    ano_fim= [ano;sexta];
```

```
case 6 %Sábado
```

```
    semana = [sabado;domingo;segunda;terca;quarta;quinta;sexta];
```

```
    ano=repmat(semana,52,1);
```

```
    ano_fim= [ano;sabado];
```

```
case 7 %Domingo
```

```
    semana = [domingo;segunda;terca;quarta;quinta;sexta;sabado];
```

```
        ano= repmat(semana,52,1);
        ano_fim= [ano;domingo];
end

%Feriados

ano_fim(1:24,1)={ 'Feriado'};%1 de Janeiro
ano_fim(2760:2784,1)={ 'Feriado'};%25 de Abril
ano_fim(2880:2904,1)={ 'Feriado'};%1 de Maio
ano_fim(3864:3888,1)={ 'Feriado'};%10 de Junho
ano_fim(5448:5472,1)={ 'Feriado'};%15 de Agosto
ano_fim(6672:6696,1)={ 'Feriado'};%5 de Outubro
ano_fim(7296:7320,1)={ 'Feriado'};%1 de Novembro
ano_fim(8016:8040,1)={ 'Feriado'};%1 de Dezembro
ano_fim(8208:8232,1)={ 'Feriado'};%8 de Dezembro
ano_fim(8616:8640,1)={ 'Feriado'};%25 de dezembro

end
```

```
% Função que define qual é o tipo de elemento construtivo definido na
% envolvente opaco. Na definição do espaço

% x1 - Valor de um Popup menu que corresponde a um determinado tipo de
% elemento construtivo
% x2 - Valor de um Popup menu que corresponde a um elemento construtivo
```

```
function[y1,y2,y3]=elementos_espaco(x1,x2)
```

```
load projecto.mat
```

```
global carproj
```

```
switch x1
```

```
    case 1
```

```
    case 2
```

```
        y1 = 'Parede';
```

```
        y2 = projecto(carproj).elementos.parede(x2-1).nome;
```

```
        y3 = projecto(carproj).elementos.parede(x2-1).u;
```

```
    case 3
```

```
        y1 = 'Pavimento';
```

```
        y2 = projecto(carproj).elementos.pavimento(x2-1).nome;
```

```
        y3 = projecto(carproj).elementos.pavimento(x2-1).u;
```

```
    case 4
```

```
        y1 = 'Cobertura';
```

```
        y2 = projecto(carproj).elementos.cobertura(x2-1).nome;
```

```
        y3 = projecto(carproj).elementos.cobertura(x2-1).u;
```

```
    case 5
```

```
        y1 = 'Teto';
```

```
        y2 = projecto(carproj).elementos.teto(x2-1).nome;
```

```
        y3 = projecto(carproj).elementos.teto(x2-1).u;
```

```
    case 6
```

```
        y1 = 'Janela';
```

```
        y2 = projecto(carproj).bdados.janela(x2-1).nome;
```

```
        y3 = projecto(carproj).bdados.janela(x2-1).uw;
```

```
    case 7
```

```
        y1 = 'Porta';
```

```
        y2 = projecto(carproj).bdados.porta(x2-1).nome;
```

```
        y3 = projecto(carproj).bdados.porta(x2-1).ud;
```

```
end
```

```
end
```

```

% Função que calcula as cargas térmicas de projeto

function[tabela2]=hora_projecto()

load projecto.mat

global carproj

pb = 101325;
Tb = 15;
hb = str2double(projecto(carproj).local.altitude);
p=double(altitude_pressao(pb,Tb,hb));

for k = 1:length(projecto(carproj).espaco)

T=str2double(projecto(carproj).espaco(k).temp_interior);
HR=str2double(projecto(carproj).espaco(k).hum_interior);
entalpia_int = calc_entalpia(T,HR,p);

T=str2double(projecto(carproj).local.tempDBV);
HR=str2double(projecto(carproj).espaco(k).infiltracoes.HR_extv);
entalpia_extv =calc_entalpia(T,HR,p);

T=str2double(projecto(carproj).local.tempDBI);
HR=str2double(projecto(carproj).espaco(k).infiltracoes.HR_exti);
entalpia_exti =calc_entalpia(T,HR,p);

    % Ganhos Externos

    for y = 1:length(projecto(carproj).espaco(k).ganhos_ext)

        ganhos_ext(k,y) = {str2double(projecto(carproj).espaco(k).ganhos_ext(y).quant) ✓
*str2double(projecto(carproj).espaco(k).ganhos_ext(y).area)...
        *projecto(carproj).espaco(k).ganhos_ext(y).u*(str2double(projecto(carproj). ✓
local.tempDBV) - str2double(projecto(carproj).espaco(k).temp_interior))};

    end

    %Ganhos internos
    %Elementos construtivos

    for w = 1:length(projecto(carproj).espaco(k).ganhos_int)

        ganhos_int(k,w) = {str2double(projecto(carproj).espaco(k).ganhos_int(w).quant) ✓
*str2double(projecto(carproj).espaco(k).ganhos_int(w).area)*projecto(carproj).espaco ✓
(k).ganhos_int(w).u...
        *(str2double(projecto(carproj).espaco(k).ganhos_int(w).temp) - str2double ✓
(projecto(carproj).espaco(k).temp_interior))};

    end

    % Ocupação

    carga_ocupacao_sensivel(k) = projecto(carproj).espaco(k).ocupacao.quant * ✓
str2double(projecto(carproj).espaco(k).ocupacao.csensivel);
    carga_ocupacao_latente(k) = projecto(carproj).espaco(k).ocupacao.quant * ✓
str2double(projecto(carproj).espaco(k).ocupacao.clatente);

```



```

% Diversas
carga_diversa_sensivel(k) = str2double(projecto(carproj).espaco(k).diversas.✓
sensivel);
carga_diversa_latente(k) = str2double(projecto(carproj).espaco(k).diversas.✓
latente);

% Iluminação
carga_iluminacao(k) = projecto(carproj).espaco(k).ilumina;

% Equipamentos Elétricos

carga_eletricos(k) = projecto(carproj).espaco(k).eletricos;

% Carga devida a Infiltrações

carga_infiltracoes_sensivel(k) = 1.23*str2double(projecto(carproj).espaco(k).✓
infiltracoes.caudal)*(str2double(projecto(carproj).local.tempDBV)-str2double✓
(projecto(carproj).espaco(k).temp_interior));
carga_infiltracoes_total(k) = 1.20*str2double(projecto(carproj).espaco(k).✓
infiltracoes.caudal)*(entalpia_extv-entalpia_int);
carga_infiltracoes_latente(k) = carga_infiltracoes_total(k)-✓
carga_infiltracoes_sensivel(k);

% Ganhos de calor - verão

ganhos_sensiveis(k) = carga_infiltracoes_sensivel(k)+carga_eletricos(k)✓
+carga_iluminacao(k)+carga_diversa_sensivel(k)+carga_ocupacao_sensivel(k)+sum✓
(cell2mat(ganhos_int(1,:)))+sum(cell2mat(ganhos_ext(1,:)));

ganhos_latentes(k) = carga_infiltracoes_latente(k) + carga_diversa_latente(k)✓
+carga_ocupacao_latente(k);

ganhos_totais(k)= ganhos_sensiveis(k)+ganhos_latentes(k);

% Perdas de calor - Inverno

% Ganhos Externos

for y = 1:length(projecto(carproj).espaco(k).ganhos_ext)

    ganhos_extinv(k,y) = {str2double(projecto(carproj).espaco(k).ganhos_ext(y).✓
quant)*str2double(projecto(carproj).espaco(k).ganhos_ext(y).area)...
    *projecto(carproj).espaco(k).ganhos_ext(y).u*(str2double(projecto(carproj).✓
local.tempDBI) - str2double(projecto(carproj).espaco(k).temp_interior))});

end

% Carga devida a Infiltrações

carga_infiltracoes_sensivelinv(k) = 1.23*str2double(projecto(carproj).espaco(k).✓
infiltracoes.caudal)*(str2double(projecto(carproj).local.tempDBI)-str2double✓
(projecto(carproj).espaco(k).temp_interior));
carga_infiltracoes_totalinv(k) = 1.20*str2double(projecto(carproj).espaco(k).✓
infiltracoes.caudal)*(entalpia_exti-entalpia_int);
carga_infiltracoes_latenteinv(k) = carga_infiltracoes_totalinv(k)-✓
carga_infiltracoes_sensivelinv(k);

```

```

for n=1:length(ganhos_int)

    if ganhos_int{1,n}>0

        ganhos_intinv{1,n}=0;

    else
        ganhos_intinv{1,n}=ganhos_int{1,n};
    end
end

perdas_sensiveis(k) = carga_infiltracoes_sensivelinv(k)+sum(cell2mat(ganhos_intinv(1,:)))+sum(cell2mat(ganhos_extinv(1,:)));

perdas_latentes(k) = carga_infiltracoes_latenteinv(k);

perdas_totais(k)= perdas_sensiveis(k)+perdas_latentes(k);

end

tabela2=table(ganhos_ext, ganhos_int, carga_ocupacao_sensivel', ↵
carga_ocupacao_latente', ...
    carga_diversa_sensivel', carga_diversa_latente', carga_iluminacao', ↵
carga_eletricos', carga_infiltracoes_sensivel', ...
    carga_infiltracoes_total', carga_infiltracoes_latente', ganhos_sensiveis', ↵
ganhos_latentes', ganhos_totais', ...
    ganhos_extinv, carga_infiltracoes_sensivelinv', carga_infiltracoes_totalinv', ↵
carga_infiltracoes_latenteinv', perdas_sensiveis', perdas_latentes', perdas_totais');

save tabela2.mat tabela2

projecto(carproj).relatorios_proj=load('tabela2.mat');

save projecto.mat projecto

delete tabela2.mat

end

```

```

function[A] = import_dat (filename)

delimiter = ';';
startRow = 2;

formatSpec = '%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%[\n\r]';

addpath('base_dados_originais\dat')

fileID = fopen(filename,'r');

dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter, 'HeaderLines', startRow-1, 'ReturnOnError', false);

fclose(fileID);

raw = repmat({''},length(dataArray{1}),length(dataArray)-1);
for col=1:length(dataArray)-1
    raw(1:length(dataArray{col}),col) = dataArray{col};
end
numericData = NaN(size(dataArray{1},1),size(dataArray,2));

for col=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]
    % Converts strings in the input cell array to numbers. Replaced non-numeric
    % strings with NaN.
    rawData = dataArray{col};
    for row=1:size(rawData, 1);
        % Create a regular expression to detect and remove non-numeric prefixes and
        % suffixes.
        regexstr = '(?<prefix>.*?)(?<numbers>([-]*(\d+[\.\,]*)+[\,\,]{0,1}\d*[eEdD]{0,1}
[-+]*\d*[i]{0,1})|([-]*(\d+[\.\,]*)*[\,\,]{1,1}\d+[eEdD]{0,1}[-+]*\d*[i]{0,1})) (?
<suffix>.*?);
        try
            result = regexp(rawData{row}, regexstr, 'names');
            numbers = result.numbers;

            % Detected commas in non-thousand locations.
            invalidThousandsSeparator = false;
            if any(numbers=='.' );
                thousandsRegExp = '^(\d+?(\.\d{3})*\,\,){0,1}\d*$';
                if isempty(regexp(thousandsRegExp, '.', 'once'));
                    numbers = NaN;
                    invalidThousandsSeparator = true;
                end
            end
            % Convert numeric strings to numbers.
            if ~invalidThousandsSeparator;
                numbers = strrep(numbers, '.', '');
                numbers = strrep(numbers, ',', '.');
                numbers = textscan(numbers, '%f');
                numericData(row, col) = numbers{1};
                raw{row, col} = numbers{1};
            end
        catch me
            end
    end
end

R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Find non-numeric cells

```

```

raw(R) = {NaN}; % Replace non-numeric cells

dia = cell2mat(raw(:, 1));
hora = cell2mat(raw(:, 2));
Temp = cell2mat(raw(:, 3));
U = cell2mat(raw(:, 4));
vms = cell2mat(raw(:, 5));
dir1 = cell2mat(raw(:, 6));
GWhm = cell2mat(raw(:, 7));
DWhm = cell2mat(raw(:, 8));
Gv_NWhm = cell2mat(raw(:, 9));
Gv_NEWhm = cell2mat(raw(:, 10));
Gv_EWhm = cell2mat(raw(:, 11));
Gv_SEWhm = cell2mat(raw(:, 12));
Gv_SWhm = cell2mat(raw(:, 13));
Gv_SWWhm = cell2mat(raw(:, 14));
Gv_WWhm = cell2mat(raw(:, 15));
Gv_NWWhm = cell2mat(raw(:, 16));

A = [dia,hora,Temp,U,vms,dir1];
clearvars filename delimiter startRow formatSpec fileID dataArray ans row col↵
numericData rawData row regexstr result numbers invalidThousandsSeparator↵
thousandsRegExp me R;

```

```
function [A] = import_epw(filename)
%% Import data from text file.
% Script for importing data from the following text file:
%
% C:\Users\João Gonçalves\Desktop\06-05-2016\Programa
% (final)\base_dados_originais\epw\Evora 221m.epw
%
% To extend the code to different selected data or a different text file,
% generate a function instead of a script.

% Auto-generated by MATLAB on 2016/05/09 11:24:56

%% Initialize variables.

delimiter = {' ',' '};

%% Read columns of data as strings:
% For more information, see the TEXTSCAN documentation.
formatSpec = '%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s'↵
's%^n\r';

%% Open the text file.
addpath('base_dados_originais\epw')
fileID = fopen(filename,'r');

%% Read columns of data according to format string.
% This call is based on the structure of the file used to generate this
% code. If an error occurs for a different file, try regenerating the code
% from the Import Tool.
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter,↵
'MultipleDelimsAsOne', true, 'ReturnOnError', false);

%% Close the text file.
fclose(fileID);

%% Convert the contents of columns containing numeric strings to numbers.
% Replace non-numeric strings with NaN.
raw = repmat({''},length(dataArray{1}),length(dataArray)-1);
for col=1:length(dataArray)-1
    raw(1:length(dataArray{col}),col) = dataArray{col};
end
numericData = NaN(size(dataArray{1},1),size(dataArray,2));

for col=↵
[1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,3↵
2,33,34,35]
    % Converts strings in the input cell array to numbers. Replaced non-numeric
    % strings with NaN.
    rawData = dataArray{col};
    for row=1:size(rawData, 1);
        % Create a regular expression to detect and remove non-numeric prefixes and
        % suffixes.
        regexstr = '(?<prefix>.*) (?<numbers>([-]*(\d+[,]*)+[\.]{0,1}\d*[eEd]{0,1}↵
[-+]*\d*[i]{0,1})|([-]*(\d+[,]*)*[\.]{1,1}\d+[eEd]{0,1}[-+]*\d*[i]{0,1})) (?↵
<suffix>.*)';
        try
            result = regexp(rawData{row}, regexstr, 'names');
            numbers = result.numbers;
```

```

% Detected commas in non-thousand locations.
invalidThousandsSeparator = false;
if any(numbers==' ');
    thousandsRegExp = '^\\d+?(\\,\\d{3})*\\.\\{0,1\\}\\d*$';
    if isempty(regexpi(thousandsRegExp, ',', 'once'));
        numbers = NaN;
        invalidThousandsSeparator = true;
    end
end
% Convert numeric strings to numbers.
if ~invalidThousandsSeparator;
    numbers = textscan(strrep(numbers, ',', ''), '%f');
    numericData(row, col) = numbers{1};
    raw{row, col} = numbers{1};
end
catch me
end
end

%% Split data into numeric and cell columns.
rawNumericColumns = raw(:, 1:
[1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,3
2,33,34,35]);
rawCellColumns = raw(:, 6);

%% Replace non-numeric cells with NaN
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x), rawNumericColumns); % Find non-
numeric cells
rawNumericColumns(R) = {NaN}; % Replace non-numeric cells

%% Allocate imported array to column variable names
LOCATION = cell2mat(rawNumericColumns(:, 1));
VarName2 = cell2mat(rawNumericColumns(:, 2));
NUTS3 = cell2mat(rawNumericColumns(:, 3));
PRT = cell2mat(rawNumericColumns(:, 4));
LNEG = cell2mat(rawNumericColumns(:, 5));
VarName6 = rawCellColumns(:, 1);
VarName7 = cell2mat(rawNumericColumns(:, 6));
VarName8 = cell2mat(rawNumericColumns(:, 7));
VarName9 = cell2mat(rawNumericColumns(:, 8));
VarName10 = cell2mat(rawNumericColumns(:, 9));
VarName11 = cell2mat(rawNumericColumns(:, 10));
VarName12 = cell2mat(rawNumericColumns(:, 11));
VarName13 = cell2mat(rawNumericColumns(:, 12));
VarName14 = cell2mat(rawNumericColumns(:, 13));
VarName15 = cell2mat(rawNumericColumns(:, 14));
VarName16 = cell2mat(rawNumericColumns(:, 15));
VarName17 = cell2mat(rawNumericColumns(:, 16));
VarName18 = cell2mat(rawNumericColumns(:, 17));
VarName19 = cell2mat(rawNumericColumns(:, 18));
VarName20 = cell2mat(rawNumericColumns(:, 19));
VarName21 = cell2mat(rawNumericColumns(:, 20));
VarName22 = cell2mat(rawNumericColumns(:, 21));
VarName23 = cell2mat(rawNumericColumns(:, 22));
VarName24 = cell2mat(rawNumericColumns(:, 23));
VarName25 = cell2mat(rawNumericColumns(:, 24));
VarName26 = cell2mat(rawNumericColumns(:, 25));
VarName27 = cell2mat(rawNumericColumns(:, 26));

```

```

VarName28 = cell2mat(rawNumericColumns(:, 27));
VarName29 = cell2mat(rawNumericColumns(:, 28));
VarName30 = cell2mat(rawNumericColumns(:, 29));
VarName31 = cell2mat(rawNumericColumns(:, 30));
VarName32 = cell2mat(rawNumericColumns(:, 31));
VarName33 = cell2mat(rawNumericColumns(:, 32));
VarName34 = cell2mat(rawNumericColumns(:, 33));
VarName35 = cell2mat(rawNumericColumns(:, 34));

```

```

A = [VarName2(9:end), NUTS3(9:end), PRT(9:end), VarName7(9:end), VarName9(9:end), ✓
VarName22(9:end), VarName21(9:end)];
%% Clear temporary variables
clearvars filename delimiter formatSpec fileID dataArray ans raw col numericData ✓
rawData row regexstr result numbers invalidThousandsSeparator thousandsRegExp me ✓
rawNumericColumns rawCellColumns R;

```

```

% Função que através do valor proveniente de Popup menus (tipo de material
% e material) estabelece o nome do material

% x1 - Valor de um Popup menu que corresponde a um determinado tipo de
% material
% x2 - Valor de um Popup menu que corresponde a um material

function [y4] = material_nome(x1,x2)

load projecto.mat

global carproj

%Água

switch x1

case 1

case 2 %Água
y4 = projecto(carproj).bdados.material.agua(x2-1).nome;

case 3 % Argamassa
y4 = projecto(carproj).bdados.material.argamassa(x2-1).nome;

case 4 %Betão
y4 = projecto(carproj).bdados.material.betao(x2-1).nome;

case 5 %Borrachas
y4 = projecto(carproj).bdados.material.borrachas(x2-1).nome;

case 6 %Fibrocimento
y4 = projecto(carproj).bdados.material.fibrocimento(x2-1).nome;

case 7 %Gases
y4 = projecto(carproj).bdados.material.gases(x2-1).nome;

case 8 %Gessos
y4 = projecto(carproj).bdados.material.gessos(x2-1).nome;

case 9 %Impermeabilização_mastiques
y4 = projecto(carproj).bdados.material.impermeabilizacao_mastiques(x2-1).nome;

case 10 % Inertes, solos e terras
y4 = projecto(carproj).bdados.material.inertes_solos_terras(x2-1).nome;

case 11 %Isolamento Térmico
y4 = projecto(carproj).bdados.material.isolamento(x2-1).nome;

case 12 %Madeira e derivados
y4 = projecto(carproj).bdados.material.madeira(x2-1).nome;

case 13 %Materiais Plásticos
y4 = projecto(carproj).bdados.material.plasticos(x2-1).nome;

case 14 % material_parede Cerâmico
y4 = projecto(carproj).bdados.material.ceramico(x2-1).nome;

case 15 %Metais
y4 = projecto(carproj).bdados.material.metais(x2-1).nome;

```



```
case 16 %Pedras
y4 = projecto(carproj).bdados.material.pedras(x2-1).nome;

case 17 %Revestimentos de pisos ou paredes
y4 = projecto(carproj).bdados.material.revestimentos_pisos_paredes(x2-1).nome;

case 18 %Vidro
y4 = projecto(carproj).bdados.material.vidros(x2-1).nome;

end
```

```

% Função que interpola/extrapola a tabela de vapor

function [ps]=myinterpl(T)

load tabela_pvapor.mat

if T>0

loc=find(T<TemperaturaC,1);
x1 = TemperaturaC(loc);
x0 = TemperaturaC(loc-1);
y1 = Pressobar(loc);
y0 = Pressobar(loc-1);

m = (y0-y1)/(x0-x1);

ps=y1+m*(T-x1);
else

loc=find(T<TemperaturaC,1);
x1 = TemperaturaC(loc+1);
x0 = TemperaturaC(loc);
y1 = Pressobar(loc+1);
y0 = Pressobar(loc);

m = (y0-y1)/(x0-x1);

ps=y1+m*(T-x1);

end

```

```

% Função que através do valor proveniente de Popup menus (tipo de material
% e material) estabelece os valores de Massa volúmica e condutibilidade
% térmica

% x1 - Valor de um Popup menu que corresponde a um determinado tipo de
% material
% x2 - Valor de um Popup menu que corresponde a um material

function [y2,y3] = rho(x1,x2)

load projecto.mat

global carproj

if x2 == 1

    y2='';
    y3='';
else

    switch x1

        case 2 %Água

            y2 = projecto(carproj).bdados.material.agua(x2-1).rho;
            y3 = projecto(carproj).bdados.material.agua(x2-1).k;

        case 3 %Argamassa

            y2 = projecto(carproj).bdados.material.argamassa(x2-1).rho;
            y3 = projecto(carproj).bdados.material.argamassa(x2-1).k;

        case 4 %Betão

            y2 = projecto(carproj).bdados.material.betao(x2-1).rho;
            y3 = projecto(carproj).bdados.material.betao(x2-1).k;

        case 5 %Borrachas

            y2 = projecto(carproj).bdados.material.borrachas(x2-1).rho;
            y3 = projecto(carproj).bdados.material.borrachas(x2-1).k;

        case 6 %Fibrocimento

            y2 = projecto(carproj).bdados.material.fibrocimento(x2-1).rho;
            y3 = projecto(carproj).bdados.material.fibrocimento(x2-1).k;

        case 7 %Gases

            y2 = projecto(carproj).bdados.material.gases(x2-1).rho;
            y3 = projecto(carproj).bdados.material.gases(x2-1).k;

        case 8 %Gessos

            y2 = projecto(carproj).bdados.material.gessos(x2-1).rho;
            y3 = projecto(carproj).bdados.material.gessos(x2-1).k;

        case 9 %Impermeabilização_mastiques

            y2 = projecto(carproj).bdados.material.impermeabilizacao_mastiques
(x2-1).rho;

```

```

(x2-1).k;

y3 = projecto(carproj).bdados.material.impermeabilizacao_mastiques✓

case 10 % Inertes, solos e terras

y2 = projecto(carproj).bdados.material.inertes_solos_terras(x2-1).✓
rho;
y3 = projecto(carproj).bdados.material.inertes_solos_terras(x2-1).k;

case 11 %Isolamento Térmico

y2 = projecto(carproj).bdados.material.isolamento(x2-1).rho;
y3 = projecto(carproj).bdados.material.isolamento(x2-1).k;

case 12 %Madeira e derivados

y2 = projecto(carproj).bdados.material.madeira(x2-1).rho;
y3 = projecto(carproj).bdados.material.madeira(x2-1).k;

case 13 %Materiais Plásticos

y2 = projecto(carproj).bdados.material.plasticos(x2-1).rho;
y3 = projecto(carproj).bdados.material.plasticos(x2-1).k;

case 14 % material_parede Cerâmico

y2 = projecto(carproj).bdados.material.ceramico(x2-1).rho;
y3 = projecto(carproj).bdados.material.ceramico(x2-1).k;

case 15 %Metais

y2 = projecto(carproj).bdados.material.metais(x2-1).rho;
y3 = projecto(carproj).bdados.material.metais(x2-1).k;

case 16 %Pedras

y2 = projecto(carproj).bdados.material.pedras(x2-1).rho;
y3 = projecto(carproj).bdados.material.pedras(x2-1).k;

case 17 %Revestimentos de pisos ou paredes

y2 = projecto(carproj).bdados.material.revestimentos_pisos_paredes✓
(x2-1).rho;
y3 = projecto(carproj).bdados.material.revestimentos_pisos_paredes✓
(x2-1).k;

case 18 %Vidro

y2 = projecto(carproj).bdados.material.vidros(x2-1).rho;
y3 = projecto(carproj).bdados.material.vidros(x2-1).k;

end

end

```

```

% Função que define a lista de materiais correspondente a um tipo de
% material

% x1 - Valor de um Popup menu que corresponde a um determinado tipo de
% material

function [y1] = tipo_material(x1)

load projecto.mat

global carproj

switch x1
    case 1 %(Selecione)

        y1='(Selecione)';

    case 2 % Água

        k = length(projecto(carproj).bdados.material.agua);
        y1 = {'(Selecione)',projecto(carproj).bdados.material.agua(1:k).nome};

    case 3 % Argamassa

        k = length(projecto(carproj).bdados.material.argamassa);
        y1 = {'(Selecione)',projecto(carproj).bdados.material.argamassa(1:k).nome};

    case 4 % Betão

        k = length(projecto(carproj).bdados.material.betao);
        y1 = {'(Selecione)',projecto(carproj).bdados.material.betao(1:k).nome};

    case 5 % Borrachas

        k = length(projecto(carproj).bdados.material.borrachas);
        y1 = {'(Selecione)',projecto(carproj).bdados.material.borrachas(1:k).nome};

    case 6 % Fibrocimento

        k = length(projecto(carproj).bdados.material.fibrocimento);
        y1 = {'(Selecione)',projecto(carproj).bdados.material.fibrocimento(1:k).nome};

    case 7 % Gases

        k = length(projecto(carproj).bdados.material.gases);
        y1 = {'(Selecione)',projecto(carproj).bdados.material.gases(1:k).nome};

    case 8 % Gessos

        k = length(projecto(carproj).bdados.material.gessos);
        y1 = {'(Selecione)',projecto(carproj).bdados.material.gessos(1:k).nome};

    case 9 % Impermeabilização e mastiques

        k = length(projecto(carproj).bdados.material.impermeabilizacao_mastiques);
        y1 = {'(Selecione)',projecto(carproj).bdados.material.impermeabilizacao_mastiques(1:k).nome};

    case 10 % Inertes, solos e terras

```

```

    k = length(projecto(carproj).bdados.material.inertes_solos_terras);
    y1 = {'(Selecione)',projecto(carproj).bdados.material.inertes_solos_terras(1:k)
k).nome};

case 11 % Isolamento Térmico

    k = length(projecto(carproj).bdados.material.isolamento);
    y1 = {'(Selecione)',projecto(carproj).bdados.material.isolamento(1:k).nome};

case 12 % Madeiras e derivados

    k = length(projecto(carproj).bdados.material.madeira);
    y1 = {'(Selecione)',projecto(carproj).bdados.material.madeira(1:k).nome};

case 13 % Materiais Plásticos

    k = length(projecto(carproj).bdados.material.plasticos);
    y1 = {'(Selecione)',projecto(carproj).bdados.material.plasticos(1:k).nome};

case 14 % Material Cerâmico

    k = length(projecto(carproj).bdados.material.ceramico);
    y1 = {'(Selecione)',projecto(carproj).bdados.material.ceramico(1:k).nome};

case 15 % Metais

    k = length(projecto(carproj).bdados.material.metais);
    y1 = {'(Selecione)',projecto(carproj).bdados.material.metais(1:k).nome};

case 16 % Pedras

    k = length(projecto(carproj).bdados.material.pedras);
    y1 = {'(Selecione)',projecto(carproj).bdados.material.pedras(1:k).nome};

case 17 % Revestimentos de pisos ou paredes

    k = length(projecto(carproj).bdados.material.revestimentos_pisos_paredes);
    y1 = {'(Selecione)',projecto(carproj).bdados.material.
revestimentos_pisos_paredes(1:k).nome};

case 18 % Vidro

    k = length(projecto(carproj).bdados.material.vidros);
    y1 = {'(Selecione)',projecto(carproj).bdados.material.vidros(1:k).nome};
end
end

```

```
% Função que calcula o coeficiente global de transmissão de calor dos
% elemetos construtivos

% rsi - Resistência térmica interna
% rse - Resistência térmica externa
% tabela - Tabela com os valores de Massa volúmica, Condutibilidade
% térmica e espessura

function[u]=u_elementos(rsi,rse,tabela)

x = length(tabela(:,1));

for i=1:x

    div(i)=(cell2mat(tabela(i,5))*10^-3)/cell2mat(tabela(i,4));

end

u=1/((1/rsi)+sum(div)+(1/rse));
```

```

% Função que calcula o coeficiente global de transmissão de calor do vidro
% duplo

% x1 - Resistência térmica interna
% x2 - Resistência térmica externa
% x3 - Resistência térmica entre vidros
% x4 - Tabela com os valores de condutibilidade térmica e espessura dos
% vidros
% z - Espessura [m]

function[ug]=ug_duplo(x1,x2,x3,x4)

z = str2double(x3(:,3));

    for i = 1:length(x3(:,1))

        y(i)={cell2mat(x3(i,2))/(z(i,1))};

    end

    w = sum(cell2mat(y));

    ug=1/(x1+x2+w+x4);

end

```



```

% Função que calcula o coeficiente global de transmissão de calor do vidro
% simples

% x1 - Resistência térmica interna
% x2 - Resistência térmica externa
% x3 - Tabela com os valores de condutibilidade térmica e espessura dos
% vidros
% z - Espessura [m]

function[ug]=ug_simples(x1,x2,x3)

z = str2double(x3(:,3));

for i = 1:length(x3(:,1))

y(i)={cell2mat(x3(i,2))/(z(i,1))};

end

w = sum(cell2mat(y));

ug=1/(x1+x2+w);

end

```

```

% Função que calcula o coeficiente global de transmissão de calor do vidro
% triplo

% x1 - Resistência térmica interna
% x2 - Resistência térmica externa
% x3 - Resistência entre vidros 1
% x4 - Resistência entre vidros 2
% x5 - Tabela com os valores de condutibilidade térmica e espessura dos
% vidros
% z - Espessura [m]

function[ug]=ug_triplo(x1,x2,x3,x4,x5)

z = str2double(x3(:,3));

for i = 1:length(x3(:,1))

y(i)={cell2mat(x3(i,2))/(z(i,1))};

end

w = sum(cell2mat(y));

ug=1/(x1+x2+w+x4+x5);

end

```

## **Anexo D – Cargas térmicas de aquecimento e arrefecimento (Cálculo Analítico)**



Elemento	Quantidade	U W/(m <sup>2</sup> .K)	A m <sup>2</sup>	Temperatura interior °C	Temperatura exterior °C	q W
Envolvente exterior						
Parede exterior 1	1	0,603	11	22	5	-112,700
Parede exterior 2	1	0,257	11	22	5	-48,071
Cobertura	1	0,411	25	22	5	-174,560
Pavimento	1	0,810	25	22	5	-344,250
Janela - Parede 1	1	5,100	4	22	5	-346,800
Janela - Parede 2	1	2,500	4	22	5	-170,000
Sub-total						-1196,382

Carga devido a Infiltrações de ar		
Carga sensível	-0,174	W
Carga latente	-0,085	W
Carga Total	-0,259	W

Carga Térmica de arrefecimento		
Carga sensível	-1196,556	W
Carga latente	-0,085	W
Carga Total	-1196,641	W

Elemento	Quantidade	U W/(m <sup>2</sup> .K)	A m <sup>2</sup>	Temperatura interior °C	Temperatura exterior °C	q W
<b>Envolvente exterior</b>						
Parede exterior 1	1	0,603	11	22	33	72,924
Parede exterior 2	1	0,257	11	22	33	31,105
Cobertura	1	0,411	25	22	33	112,951
Pavimento	1	0,810	25	22	33	222,750
Janela - Parede 1	1	5,100	4	22	33	224,400
Janela - Parede 2	1	2,500	4	22	33	110,000
Sub-total						774,129

Elemento	Quantidade	U W/(m <sup>2</sup> .K)	A m <sup>2</sup>	Temperatura interior °C	Temperatura adjacente °C	q W
<b>Envolvente interior</b>						
Parede interior 1	1	1,453	13,005	22	25	56,670
Parede interior 2	1	0,352	15	22	25	15,845
Porta - Parede 1	1	2,610	1,995	22	25	15,621
Sub-total						88,136

<b>Ocupação</b>		
Carga sensível	125,000	W
Carga latente	91,667	W

<b>Cargas diversas</b>		
Carga sensível	0,000	W
Carga latente	0,000	W

<b>Iluminação</b>		
Carga sensível	262,500	W

<b>Equipamentos Elétricos</b>		
Carga sensível	375,000	W

<b>Carga devido a Infiltrações de ar</b>		
Carga sensível	0,113	W
Carga latente	0,113	W
Carga Total	0,226	W

<b>Carga Térmica de arrefecimento</b>		
Carga sensível	1624,878	W
Carga latente	91,780	W
Carga Total	1716,657	W

## **Anexo E – Relatórios do programa HAP**





**Air System Information**

Air System Name .....	Sistema Escritório	Number of zones .....	1
Equipment Class .....	TERM	Floor Area .....	25,0 m²
Air System Type .....	PKG-FC	Location .....	Lisbon, Portugal

**Sizing Calculation Information**

Calculation Months .....	Jan to Dec	Zone L/s Sizing .....	Sum of space airflow rates
Sizing Data .....	Calculated	Space L/s Sizing .....	Individual peak space loads

### Air System Information

Air System Name ..... **Sistema Escritório**  
Equipment Class ..... **TERM**  
Air System Type ..... **PKG-FC**

Number of zones ..... **1**  
Floor Area ..... **25,0** m²  
Location ..... **Lisbon, Portugal**

### Sizing Calculation Information

Calculation Months ..... **Jan to Dec**  
Sizing Data ..... **Calculated**

Zone L/s Sizing ..... **Sum of space airflow rates**  
Space L/s Sizing ..... **Individual peak space loads**

### Zone Sizing Data

Zone Name	Maximum Cooling Sensible (kW)	Design Airflow (L/s)	Minimum Airflow (L/s)	Time of Peak Load	Maximum Heating Load (kW)	Zone Floor Area (m²)	Zone L/(s-m²)
Zone 1	2,8	315	315	Jul 1400	1,1	25,0	12,59

### Terminal Unit Sizing Data - Cooling

Zone Name	Total Coil Load (kW)	Sens Coil Load (kW)	Coil Entering DB / WB (°C)	Coil Leaving DB / WB (°C)	Water Flow @ 5,6 °K (L/s)	Time of Peak Load
Zone 1	0,0	0,0	-18,3 / -18,3	-18,3 / -18,3	0,00	Des 0000

### Terminal Unit Sizing Data - Heating, Fan, Ventilation

Zone Name	Heating Coil Load (kW)	Heating Coil Ent/Lvg DB (°C)	Htg Coil Water Flow @11,1 °K (L/s)	Fan Design Airflow (L/s)	Fan Motor (BHP)	Fan Motor (kW)	OA Vent Design Airflow (L/s)
Zone 1	0,0	-18,3 / -18,3	0,00	315	0,000	0,000	12

### Space Loads and Airflows

Zone Name / Space Name	Mult.	Cooling Sensible (kW)	Time of Load	Air Flow (L/s)	Heating Load (kW)	Floor Area (m²)	Space L/(s-m²)
<b>Zone 1</b>							
Escritório	1	2,8	Jul 1400	315	1,1	25,0	12,59

	DESIGN COOLING			DESIGN HEATING		
	NO COOLING DATA NO COOLING OA DB / WB			HEATING DATA AT DES HTG HEATING OA DB / WB 5,0 °C / 4,3 °C		
ZONE LOADS	Details	Sensible (W)	Latent (W)	Details	Sensible (W)	Latent (W)
Window & Skylight Solar Loads	8 m²	-	-	8 m²	-	-
Wall Transmission	22 m²	-	-	22 m²	151	-
Roof Transmission	25 m²	-	-	25 m²	164	-
Window Transmission	8 m²	-	-	8 m²	486	-
Skylight Transmission	0 m²	-	-	0 m²	0	-
Door Loads	0 m²	-	-	0 m²	0	-
Floor Transmission	25 m²	-	-	25 m²	324	-
Partitions	28 m²	-	-	28 m²	0	-
Ceiling	0 m²	-	-	0 m²	0	-
Overhead Lighting	-	-	-	0	0	-
Task Lighting	-	-	-	0	0	-
Electric Equipment	-	-	-	0	0	-
People	-	-	-	0	0	0
Infiltration	-	-	-	-	0	0
Miscellaneous	-	-	-	-	0	0
Safety Factor	0% / 0%	-	-	0%	0	0
>> Total Zone Loads	-	-	-	-	1126	0
Zone Conditioning	-	-	-	-	-60	0
Plenum Wall Load	0%	-	-	0	0	-
Plenum Roof Load	0%	-	-	0	0	-
Plenum Lighting Load	0%	-	-	0	0	-
Exhaust Fan Load	-	-	-	0 L/s	0	-
Ventilation Load	-	-	-	12 L/s	70	0
Ventilation Fan Load	-	-	-	0 L/s	0	-
Space Fan Coil Fans	-	-	-	-	0	-
Duct Heat Gain / Loss	0%	-	-	0%	0	-
>> Total System Loads	-	-	-	-	10	0
Terminal Unit Cooling	-	-	-	-	0	0
Terminal Unit Heating	-	-	-	-	0	-
>> Total Conditioning	-	-	-	-	0	0
Key:	Positive values are clg loads Negative values are htg loads			Positive values are htg loads Negative values are clg loads		

Zone 1	DESIGN COOLING			DESIGN HEATING		
	COOLING DATA AT Jul 1400			HEATING DATA AT DES HTG		
	COOLING OA DB / WB 32,7 °C / 22,2 °C			HEATING OA DB / WB 5,0 °C / 4,3 °C		
	OCCUPIED T-STAT 22,0 °C			OCCUPIED T-STAT 21,0 °C		
ZONE LOADS	Details	Sensible (W)	Latent (W)	Details	Sensible (W)	Latent (W)
Window & Skylight Solar Loads	8 m²	1390	-	8 m²	-	-
Wall Transmission	22 m²	74	-	22 m²	151	-
Roof Transmission	25 m²	84	-	25 m²	164	-
Window Transmission	8 m²	270	-	8 m²	486	-
Skylight Transmission	0 m²	0	-	0 m²	0	-
Door Loads	0 m²	0	-	0 m²	0	-
Floor Transmission	25 m²	180	-	25 m²	324	-
Partitions	28 m²	73	-	28 m²	0	-
Ceiling	0 m²	0	-	0 m²	0	-
Overhead Lighting	263 W	262	-	0	0	-
Task Lighting	0 W	0	-	0	0	-
Electric Equipment	375 W	375	-	0	0	-
People	2	125	92	0	0	0
Infiltration	-	0	0	-	0	0
Miscellaneous	-	0	0	-	0	0
Safety Factor	0% / 0%	0	0	0%	0	0
>> Total Zone Loads	-	2833	92	-	1126	0

TABLE 1.1.A. COMPONENT LOADS FOR SPACE "Escritório" IN ZONE "Zone 1"						
	DESIGN COOLING			DESIGN HEATING		
	COOLING DATA AT Jul 1400 COOLING OA DB / WB 32,7 °C / 22,2 °C OCCUPIED T-STAT 22,0 °C			HEATING DATA AT DES HTG HEATING OA DB / WB 5,0 °C / 4,3 °C OCCUPIED T-STAT 21,0 °C		
		Sensible	Latent		Sensible	Latent
SPACE LOADS	Details	(W)	(W)	Details	(W)	(W)
Window & Skylight Solar Loads	8 m²	1390	-	8 m²	-	-
Wall Transmission	22 m²	74	-	22 m²	151	-
Roof Transmission	25 m²	84	-	25 m²	164	-
Window Transmission	8 m²	270	-	8 m²	486	-
Skylight Transmission	0 m²	0	-	0 m²	0	-
Door Loads	0 m²	0	-	0 m²	0	-
Floor Transmission	25 m²	180	-	25 m²	324	-
Partitions	28 m²	73	-	28 m²	0	-
Ceiling	0 m²	0	-	0 m²	0	-
Overhead Lighting	263 W	262	-	0	0	-
Task Lighting	0 W	0	-	0	0	-
Electric Equipment	375 W	375	-	0	0	-
People	2	125	92	0	0	0
Infiltration	-	0	0	-	0	0
Miscellaneous	-	0	0	-	0	0
Safety Factor	0% / 0%	0	0	0%	0	0
>> Total Zone Loads	-	2833	92	-	1126	0

TABLE 1.1.B. ENVELOPE LOADS FOR SPACE "Escritório" IN ZONE "Zone 1"						
	Area	U-Value	Shade	COOLING	COOLING	HEATING
	(m²)	(W/(m²·°K))	Coeff.	TRANS	SOLAR	TRANS
				(W)	(W)	(W)
<b>N EXPOSURE</b>						
WALL	11	0,603	-	46	-	106
WINDOW 1	4	5,100	1,000	181	561	326
<b>E EXPOSURE</b>						
WALL	11	0,257	-	28	-	45
WINDOW 1	4	2,500	1,000	89	829	160
<b>H EXPOSURE</b>						
ROOF	25	0,411	-	84	-	164

ZONE: Zone 1 DESIGN MONTH: JANUARY									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	19,2	32,0	42	314,7	981,3	171,3	0,0	0,0	0,0
0100	18,7	31,8	43	314,7	948,7	174,1	0,0	0,0	0,0
0200	18,2	31,6	43	314,7	915,6	178,3	0,0	0,0	0,0
0300	17,8	31,4	44	314,7	885,6	181,2	0,0	0,0	0,0
0400	17,5	31,2	44	314,7	858,7	182,8	0,0	0,0	0,0
0500	17,4	31,1	45	314,7	838,5	182,2	0,0	0,0	0,0
0600	17,6	30,9	45	314,7	828,8	178,2	0,0	0,0	0,0
0700	18,1	30,8	45	314,7	829,8	170,9	0,0	0,0	0,0
0800	19,0	31,1	45	314,7	947,9	158,6	0,0	0,0	0,0
0900	20,3	31,6	43	314,7	1147,6	165,1	0,0	0,0	0,0
1000	21,8	32,0	43	314,7	1270,3	148,1	0,0	0,0	0,0
1100	23,5	32,3	42	314,7	1344,6	120,7	0,0	0,0	0,0
1200	25,1	32,6	41	314,7	1388,9	95,6	0,0	0,0	0,0
1300	26,3	32,8	41	314,7	1446,6	92,1	0,0	0,0	0,0
1400	27,1	33,0	40	314,7	1476,7	78,6	0,0	0,0	0,0
1500	27,4	33,1	40	314,7	1468,0	72,8	0,0	0,0	0,0
1600	27,1	33,0	40	314,7	1414,7	75,7	0,0	0,0	0,0
1700	26,4	32,9	41	314,7	1332,9	80,0	0,0	0,0	0,0
1800	25,3	32,7	41	314,7	1278,8	93,0	0,0	0,0	0,0
1900	24,0	32,5	41	314,7	1219,2	112,5	0,0	0,0	0,0
2000	22,7	32,3	42	314,7	1160,2	127,2	0,0	0,0	0,0
2100	21,6	32,1	42	314,7	1108,1	139,4	0,0	0,0	0,0
2200	20,6	31,9	43	314,7	1059,4	150,1	0,0	0,0	0,0
2300	19,8	31,7	43	314,7	1017,1	158,4	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: FEBRUARY									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	20,3	33,4	40	314,7	1112,5	174,9	0,0	0,0	0,0
0100	19,8	33,2	40	314,7	1077,0	177,5	0,0	0,0	0,0
0200	19,3	33,0	41	314,7	1041,1	181,5	0,0	0,0	0,0
0300	18,9	32,7	41	314,7	1008,4	184,1	0,0	0,0	0,0
0400	18,6	32,5	42	314,7	979,0	185,6	0,0	0,0	0,0
0500	18,5	32,4	42	314,7	956,3	184,8	0,0	0,0	0,0
0600	18,7	32,2	43	314,7	944,2	180,6	0,0	0,0	0,0
0700	19,2	32,1	43	314,7	943,0	173,1	0,0	0,0	0,0
0800	20,1	32,6	42	314,7	1157,3	177,6	0,0	0,0	0,0
0900	21,4	33,0	41	314,7	1314,9	165,0	0,0	0,0	0,0
1000	22,9	33,4	40	314,7	1427,3	143,5	0,0	0,0	0,0
1100	24,6	33,7	39	314,7	1501,1	118,8	0,0	0,0	0,0
1200	26,2	34,0	39	314,7	1545,4	101,6	0,0	0,0	0,0
1300	27,4	34,2	38	314,7	1603,0	95,2	0,0	0,0	0,0
1400	28,2	34,5	38	314,7	1636,7	82,3	0,0	0,0	0,0
1500	28,5	34,6	38	314,7	1634,4	77,8	0,0	0,0	0,0
1600	28,2	34,5	38	314,7	1590,5	81,1	0,0	0,0	0,0
1700	27,5	34,4	38	314,7	1511,3	85,3	0,0	0,0	0,0
1800	26,4	34,1	39	314,7	1430,9	101,6	0,0	0,0	0,0
1900	25,1	33,9	39	314,7	1367,2	116,6	0,0	0,0	0,0
2000	23,8	33,7	40	314,7	1304,4	131,0	0,0	0,0	0,0
2100	22,7	33,5	40	314,7	1248,8	143,0	0,0	0,0	0,0
2200	21,7	33,3	41	314,7	1196,7	153,6	0,0	0,0	0,0
2300	20,9	33,1	41	314,7	1151,3	161,8	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: MARCH									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	22,0	36,2	36	314,7	1376,8	190,4	0,0	0,0	0,0
0100	21,5	36,0	37	314,7	1333,1	192,3	0,0	0,0	0,0
0200	21,0	35,7	37	314,7	1289,4	195,8	0,0	0,0	0,0
0300	20,6	35,5	38	314,7	1249,1	197,9	0,0	0,0	0,0
0400	20,3	35,2	38	314,7	1212,4	198,8	0,0	0,0	0,0
0500	20,2	35,0	39	314,7	1182,8	197,5	0,0	0,0	0,0
0600	20,4	34,8	39	314,7	1164,1	192,8	0,0	0,0	0,0
0700	20,9	35,2	39	314,7	1330,4	188,4	0,0	0,0	0,0
0800	21,8	35,7	38	314,7	1534,8	186,5	0,0	0,0	0,0
0900	23,1	36,1	37	314,7	1688,5	170,9	0,0	0,0	0,0
1000	24,6	36,5	36	314,7	1802,3	158,1	0,0	0,0	0,0
1100	26,3	36,8	35	314,7	1873,5	137,0	0,0	0,0	0,0
1200	27,9	37,1	35	314,7	1917,9	122,4	0,0	0,0	0,0
1300	29,1	37,3	34	314,7	1980,6	110,4	0,0	0,0	0,0
1400	29,9	37,6	34	314,7	2014,9	100,0	0,0	0,0	0,0
1500	30,2	37,7	34	314,7	2010,5	95,6	0,0	0,0	0,0
1600	29,9	37,6	34	314,7	1961,6	100,0	0,0	0,0	0,0
1700	29,2	37,5	34	314,7	1874,7	109,5	0,0	0,0	0,0
1800	28,1	37,1	35	314,7	1755,9	118,8	0,0	0,0	0,0
1900	26,8	36,9	35	314,7	1678,7	133,7	0,0	0,0	0,0
2000	25,5	36,6	36	314,7	1605,3	147,7	0,0	0,0	0,0
2100	24,4	36,4	36	314,7	1539,8	159,3	0,0	0,0	0,0
2200	23,4	36,2	37	314,7	1478,4	169,4	0,0	0,0	0,0
2300	22,6	35,9	37	314,7	1424,2	177,0	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: APRIL									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	22,5	39,7	31	314,7	1686,2	229,4	0,0	0,0	0,0
0100	22,0	39,4	32	314,7	1627,4	235,5	0,0	0,0	0,0
0200	21,5	39,0	32	314,7	1569,3	238,0	0,0	0,0	0,0
0300	21,1	38,7	33	314,7	1515,2	239,0	0,0	0,0	0,0
0400	20,8	38,4	33	314,7	1465,2	238,8	0,0	0,0	0,0
0500	20,7	38,1	34	314,7	1423,0	236,3	0,0	0,0	0,0
0600	20,9	38,2	34	314,7	1517,9	233,5	0,0	0,0	0,0
0700	21,4	38,8	33	314,7	1791,9	232,0	0,0	0,0	0,0
0800	22,3	39,3	32	314,7	2002,0	226,2	0,0	0,0	0,0
0900	23,6	39,8	31	314,7	2172,1	219,7	0,0	0,0	0,0
1000	25,1	40,3	30	314,7	2294,9	203,7	0,0	0,0	0,0
1100	26,8	40,6	30	314,7	2364,6	182,1	0,0	0,0	0,0
1200	28,4	40,9	29	314,7	2415,5	164,1	0,0	0,0	0,0
1300	29,6	41,2	29	314,7	2486,1	155,8	0,0	0,0	0,0
1400	30,4	41,5	29	314,7	2521,8	145,4	0,0	0,0	0,0
1500	30,7	41,6	28	314,7	2512,0	143,8	0,0	0,0	0,0
1600	30,4	41,5	28	314,7	2452,5	148,8	0,0	0,0	0,0
1700	29,7	41,3	29	314,7	2349,0	153,8	0,0	0,0	0,0
1800	28,6	41,0	29	314,7	2208,7	162,2	0,0	0,0	0,0
1900	27,3	40,6	30	314,7	2074,9	174,0	0,0	0,0	0,0
2000	26,0	40,2	31	314,7	1982,0	188,0	0,0	0,0	0,0
2100	24,9	39,9	31	314,7	1898,3	198,7	0,0	0,0	0,0
2200	23,9	39,6	32	314,7	1819,8	207,9	0,0	0,0	0,0
2300	23,1	39,3	32	314,7	1749,2	214,7	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: MAY									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	23,1	40,2	31	314,7	1735,1	227,7	0,0	0,0	0,0
0100	22,6	39,8	32	314,7	1676,0	233,8	0,0	0,0	0,0
0200	22,1	39,5	33	314,7	1617,5	236,2	0,0	0,0	0,0
0300	21,7	39,2	33	314,7	1563,0	237,3	0,0	0,0	0,0
0400	21,4	38,9	34	314,7	1512,8	237,1	0,0	0,0	0,0
0500	21,3	38,6	34	314,7	1470,2	234,5	0,0	0,0	0,0
0600	21,5	39,0	34	314,7	1697,0	237,7	0,0	0,0	0,0
0700	22,0	39,5	33	314,7	1891,4	232,1	0,0	0,0	0,0
0800	22,9	39,9	32	314,7	2055,8	225,3	0,0	0,0	0,0
0900	24,2	40,3	31	314,7	2203,0	215,2	0,0	0,0	0,0
1000	25,7	40,7	31	314,7	2311,3	199,2	0,0	0,0	0,0
1100	27,4	41,0	30	314,7	2379,1	178,9	0,0	0,0	0,0
1200	29,0	41,3	30	314,7	2435,3	162,0	0,0	0,0	0,0
1300	30,2	41,6	29	314,7	2502,8	152,6	0,0	0,0	0,0
1400	31,0	41,9	29	314,7	2538,9	142,5	0,0	0,0	0,0
1500	31,3	42,0	29	314,7	2535,4	141,4	0,0	0,0	0,0
1600	31,0	42,0	29	314,7	2486,4	141,7	0,0	0,0	0,0
1700	30,3	41,8	29	314,7	2407,4	152,8	0,0	0,0	0,0
1800	29,2	41,5	29	314,7	2301,2	163,0	0,0	0,0	0,0
1900	27,9	41,0	30	314,7	2127,6	177,0	0,0	0,0	0,0
2000	26,6	40,7	31	314,7	2033,1	186,5	0,0	0,0	0,0
2100	25,5	40,4	31	314,7	1948,7	197,1	0,0	0,0	0,0
2200	24,5	40,1	32	314,7	1869,5	206,3	0,0	0,0	0,0
2300	23,7	39,8	32	314,7	1798,5	213,0	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: JUNE									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	24,2	41,4	30	314,7	1853,6	228,3	0,0	0,0	0,0
0100	23,7	41,0	31	314,7	1792,3	234,3	0,0	0,0	0,0
0200	23,2	40,7	31	314,7	1731,7	236,6	0,0	0,0	0,0
0300	22,8	40,3	32	314,7	1675,2	237,6	0,0	0,0	0,0
0400	22,5	40,0	33	314,7	1623,0	237,3	0,0	0,0	0,0
0500	22,4	39,7	33	314,7	1582,8	234,8	0,0	0,0	0,0
0600	22,6	40,2	32	314,7	1828,0	235,4	0,0	0,0	0,0
0700	23,1	40,6	32	314,7	2016,1	237,4	0,0	0,0	0,0
0800	24,0	41,0	31	314,7	2169,9	230,4	0,0	0,0	0,0
0900	25,3	41,4	30	314,7	2315,3	215,6	0,0	0,0	0,0
1000	26,8	41,8	30	314,7	2426,1	199,7	0,0	0,0	0,0
1100	28,5	42,2	29	314,7	2496,8	179,6	0,0	0,0	0,0
1200	30,1	42,4	29	314,7	2551,9	162,2	0,0	0,0	0,0
1300	31,3	42,8	28	314,7	2619,4	152,9	0,0	0,0	0,0
1400	32,1	43,0	28	314,7	2658,9	143,2	0,0	0,0	0,0
1500	32,4	43,1	28	314,7	2658,8	142,2	0,0	0,0	0,0
1600	32,1	43,1	28	314,7	2614,4	142,7	0,0	0,0	0,0
1700	31,4	43,0	28	314,7	2547,4	154,4	0,0	0,0	0,0
1800	30,3	42,8	28	314,7	2452,9	165,1	0,0	0,0	0,0
1900	29,0	42,3	29	314,7	2281,9	179,0	0,0	0,0	0,0
2000	27,7	41,9	30	314,7	2162,2	186,9	0,0	0,0	0,0
2100	26,6	41,6	30	314,7	2074,8	198,0	0,0	0,0	0,0
2200	25,6	41,3	31	314,7	1992,9	207,1	0,0	0,0	0,0
2300	24,8	41,0	31	314,7	1919,3	213,8	0,0	0,0	0,0



ZONE: Zone 1 DESIGN MONTH: JULY									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	24,8	42,5	29	314,7	1961,6	236,3	0,0	0,0	0,0
0100	24,3	42,1	29	314,7	1896,4	242,2	0,0	0,0	0,0
0200	23,8	41,8	30	314,7	1832,2	244,3	0,0	0,0	0,0
0300	23,4	41,4	30	314,7	1772,2	245,0	0,0	0,0	0,0
0400	23,1	41,1	31	314,7	1716,6	244,5	0,0	0,0	0,0
0500	23,0	40,8	32	314,7	1669,0	241,5	0,0	0,0	0,0
0600	23,2	41,1	31	314,7	1878,8	242,5	0,0	0,0	0,0
0700	23,7	41,7	30	314,7	2110,9	242,3	0,0	0,0	0,0
0800	24,6	42,1	29	314,7	2295,8	235,1	0,0	0,0	0,0
0900	25,9	42,6	29	314,7	2464,2	225,9	0,0	0,0	0,0
1000	27,4	43,1	28	314,7	2589,2	210,2	0,0	0,0	0,0
1100	29,1	43,4	27	314,7	2667,7	189,7	0,0	0,0	0,0
1200	30,7	43,7	27	314,7	2720,4	171,2	0,0	0,0	0,0
1300	31,9	44,0	27	314,7	2792,8	162,8	0,0	0,0	0,0
1400	32,7	44,3	26	314,7	2833,3	153,0	0,0	0,0	0,0
1500	33,0	44,4	26	314,7	2832,9	152,3	0,0	0,0	0,0
1600	32,7	44,4	26	314,7	2783,8	157,8	0,0	0,0	0,0
1700	32,0	44,3	26	314,7	2699,9	163,8	0,0	0,0	0,0
1800	30,9	44,0	27	314,7	2590,2	173,8	0,0	0,0	0,0
1900	29,6	43,5	27	314,7	2401,8	187,4	0,0	0,0	0,0
2000	28,3	43,1	28	314,7	2287,0	200,8	0,0	0,0	0,0
2100	27,2	42,8	28	314,7	2195,1	206,4	0,0	0,0	0,0
2200	26,2	42,4	29	314,7	2108,9	220,4	0,0	0,0	0,0
2300	25,4	42,1	29	314,7	2031,3	226,9	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: AUGUST									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	24,8	41,5	30	314,7	1869,0	222,8	0,0	0,0	0,0
0100	24,3	41,1	31	314,7	1808,9	228,8	0,0	0,0	0,0
0200	23,8	40,8	31	314,7	1749,5	231,2	0,0	0,0	0,0
0300	23,4	40,5	32	314,7	1694,2	232,2	0,0	0,0	0,0
0400	23,1	40,2	33	314,7	1643,1	232,0	0,0	0,0	0,0
0500	23,0	39,9	33	314,7	1599,7	224,3	0,0	0,0	0,0
0600	23,2	39,9	33	314,7	1667,8	224,9	0,0	0,0	0,0
0700	23,7	40,5	32	314,7	1949,0	229,6	0,0	0,0	0,0
0800	24,6	41,1	31	314,7	2168,1	223,4	0,0	0,0	0,0
0900	25,9	41,6	30	314,7	2346,3	211,4	0,0	0,0	0,0
1000	27,4	42,0	29	314,7	2477,3	193,4	0,0	0,0	0,0
1100	29,1	42,4	29	314,7	2554,3	176,9	0,0	0,0	0,0
1200	30,7	42,7	28	314,7	2606,0	158,7	0,0	0,0	0,0
1300	31,9	43,0	28	314,7	2678,4	146,5	0,0	0,0	0,0
1400	32,7	43,3	28	314,7	2717,6	140,1	0,0	0,0	0,0
1500	33,0	43,4	27	314,7	2710,8	138,4	0,0	0,0	0,0
1600	32,7	43,4	27	314,7	2653,7	138,1	0,0	0,0	0,0
1700	32,0	43,2	28	314,7	2551,3	148,1	0,0	0,0	0,0
1800	30,9	42,8	28	314,7	2409,5	156,5	0,0	0,0	0,0
1900	29,6	42,4	29	314,7	2265,6	167,8	0,0	0,0	0,0
2000	28,3	42,1	29	314,7	2170,9	182,0	0,0	0,0	0,0
2100	27,2	41,8	30	314,7	2085,5	192,7	0,0	0,0	0,0
2200	26,2	41,4	30	314,7	2005,4	201,8	0,0	0,0	0,0
2300	25,4	41,1	31	314,7	1933,4	208,5	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: SEPTEMBER									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	23,7	38,5	36	314,7	1589,2	197,7	0,0	0,0	0,0
0100	23,2	38,2	36	314,7	1540,1	199,2	0,0	0,0	0,0
0200	22,7	37,9	37	314,7	1491,3	202,3	0,0	0,0	0,0
0300	22,3	37,7	37	314,7	1446,2	204,0	0,0	0,0	0,0
0400	22,0	37,4	38	314,7	1404,8	204,6	0,0	0,0	0,0
0500	21,9	37,1	38	314,7	1370,9	202,9	0,0	0,0	0,0
0600	22,1	36,9	38	314,7	1348,0	197,8	0,0	0,0	0,0
0700	22,6	37,5	37	314,7	1600,1	200,8	0,0	0,0	0,0
0800	23,5	38,0	36	314,7	1822,3	204,1	0,0	0,0	0,0
0900	24,8	38,5	35	314,7	1989,9	188,7	0,0	0,0	0,0
1000	26,3	38,9	34	314,7	2105,7	169,4	0,0	0,0	0,0
1100	28,0	39,2	34	314,7	2169,5	150,4	0,0	0,0	0,0
1200	29,6	39,5	33	314,7	2224,1	129,4	0,0	0,0	0,0
1300	30,8	39,9	33	314,7	2290,0	118,6	0,0	0,0	0,0
1400	31,6	40,1	32	314,7	2319,7	107,9	0,0	0,0	0,0
1500	31,9	40,2	32	314,7	2304,1	109,5	0,0	0,0	0,0
1600	31,6	40,1	32	314,7	2237,9	108,9	0,0	0,0	0,0
1700	30,9	39,8	33	314,7	2127,4	117,9	0,0	0,0	0,0
1800	29,8	39,5	33	314,7	2002,8	127,4	0,0	0,0	0,0
1900	28,5	39,3	34	314,7	1920,8	142,1	0,0	0,0	0,0
2000	27,2	39,0	34	314,7	1841,0	155,8	0,0	0,0	0,0
2100	26,1	38,7	35	314,7	1769,3	167,1	0,0	0,0	0,0
2200	25,1	38,4	35	314,7	1702,0	176,8	0,0	0,0	0,0
2300	24,3	38,2	36	314,7	1642,0	184,1	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: OCTOBER									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	22,6	36,2	39	314,7	1377,1	182,4	0,0	0,0	0,0
0100	22,1	36,0	40	314,7	1335,4	184,5	0,0	0,0	0,0
0200	21,6	35,7	40	314,7	1293,7	188,0	0,0	0,0	0,0
0300	21,2	35,5	41	314,7	1255,3	190,3	0,0	0,0	0,0
0400	20,9	35,3	41	314,7	1220,5	191,3	0,0	0,0	0,0
0500	20,8	35,1	42	314,7	1192,8	190,1	0,0	0,0	0,0
0600	21,0	34,9	42	314,7	1175,9	185,5	0,0	0,0	0,0
0700	21,5	35,1	42	314,7	1305,8	183,1	0,0	0,0	0,0
0800	22,4	35,7	40	314,7	1552,3	194,3	0,0	0,0	0,0
0900	23,7	36,2	39	314,7	1715,9	180,8	0,0	0,0	0,0
1000	25,2	36,6	38	314,7	1821,7	152,4	0,0	0,0	0,0
1100	26,9	36,8	38	314,7	1877,1	133,4	0,0	0,0	0,0
1200	28,5	37,1	37	314,7	1935,6	112,1	0,0	0,0	0,0
1300	29,7	37,4	37	314,7	1994,9	105,0	0,0	0,0	0,0
1400	30,5	37,6	36	314,7	2016,8	92,0	0,0	0,0	0,0
1500	30,8	37,7	36	314,7	1993,3	86,1	0,0	0,0	0,0
1600	30,5	37,6	36	314,7	1919,2	89,6	0,0	0,0	0,0
1700	29,8	37,3	37	314,7	1807,3	98,5	0,0	0,0	0,0
1800	28,7	37,1	37	314,7	1737,6	111,5	0,0	0,0	0,0
1900	27,4	36,9	38	314,7	1666,1	125,6	0,0	0,0	0,0
2000	26,1	36,6	38	314,7	1595,9	139,6	0,0	0,0	0,0
2100	25,0	36,4	39	314,7	1533,2	151,3	0,0	0,0	0,0
2200	24,0	36,2	39	314,7	1474,3	161,4	0,0	0,0	0,0
2300	23,2	35,9	40	314,7	1422,3	169,2	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: NOVEMBER									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	20,9	33,6	45	314,7	1129,6	169,9	0,0	0,0	0,0
0100	20,4	33,4	46	314,7	1095,0	172,6	0,0	0,0	0,0
0200	19,9	33,1	46	314,7	1060,2	176,6	0,0	0,0	0,0
0300	19,5	32,9	47	314,7	1028,4	179,3	0,0	0,0	0,0
0400	19,2	32,7	47	314,7	999,9	180,8	0,0	0,0	0,0
0500	19,1	32,6	47	314,7	978,2	180,1	0,0	0,0	0,0
0600	19,3	32,4	48	314,7	967,1	176,0	0,0	0,0	0,0
0700	19,8	32,3	48	314,7	966,7	168,5	0,0	0,0	0,0
0800	20,7	32,8	47	314,7	1189,0	174,5	0,0	0,0	0,0
0900	22,0	33,3	45	314,7	1364,3	165,5	0,0	0,0	0,0
1000	23,5	33,7	44	314,7	1471,1	142,6	0,0	0,0	0,0
1100	25,2	34,0	44	314,7	1528,8	116,9	0,0	0,0	0,0
1200	26,8	34,2	43	314,7	1584,5	100,1	0,0	0,0	0,0
1300	28,0	34,5	42	314,7	1640,7	94,7	0,0	0,0	0,0
1400	28,8	34,7	42	314,7	1660,9	75,9	0,0	0,0	0,0
1500	29,1	34,8	42	314,7	1637,5	70,6	0,0	0,0	0,0
1600	28,8	34,6	42	314,7	1565,7	73,5	0,0	0,0	0,0
1700	28,1	34,5	42	314,7	1495,6	79,8	0,0	0,0	0,0
1800	27,0	34,3	43	314,7	1439,5	92,1	0,0	0,0	0,0
1900	25,7	34,1	43	314,7	1377,7	111,6	0,0	0,0	0,0
2000	24,4	33,9	44	314,7	1316,6	126,1	0,0	0,0	0,0
2100	23,3	33,7	44	314,7	1262,4	138,2	0,0	0,0	0,0
2200	22,3	33,5	45	314,7	1211,6	148,8	0,0	0,0	0,0
2300	21,5	33,3	45	314,7	1167,3	157,0	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: DECEMBER									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	19,8	32,3	45	314,7	1005,8	167,4	0,0	0,0	0,0
0100	19,3	32,1	46	314,7	973,8	170,3	0,0	0,0	0,0
0200	18,8	31,9	46	314,7	941,5	174,5	0,0	0,0	0,0
0300	18,4	31,7	47	314,7	912,2	177,4	0,0	0,0	0,0
0400	18,1	31,5	47	314,7	886,0	179,0	0,0	0,0	0,0
0500	18,0	31,3	48	314,7	866,5	178,5	0,0	0,0	0,0
0600	18,2	31,2	48	314,7	857,3	174,6	0,0	0,0	0,0
0700	18,7	31,1	48	314,7	859,0	167,2	0,0	0,0	0,0
0800	19,6	31,4	47	314,7	971,4	154,5	0,0	0,0	0,0
0900	20,9	31,9	46	314,7	1179,3	158,8	0,0	0,0	0,0
1000	22,4	32,3	45	314,7	1299,4	144,1	0,0	0,0	0,0
1100	24,1	32,6	44	314,7	1365,9	114,9	0,0	0,0	0,0
1200	25,7	32,8	44	314,7	1413,6	94,2	0,0	0,0	0,0
1300	26,9	33,1	43	314,7	1471,3	91,2	0,0	0,0	0,0
1400	27,7	33,3	42	314,7	1495,3	74,9	0,0	0,0	0,0
1500	28,0	33,3	42	314,7	1477,2	67,7	0,0	0,0	0,0
1600	27,7	33,2	42	314,7	1412,1	70,1	0,0	0,0	0,0
1700	27,0	33,1	43	314,7	1349,9	76,8	0,0	0,0	0,0
1800	25,9	33,0	43	314,7	1297,5	89,0	0,0	0,0	0,0
1900	24,6	32,8	44	314,7	1239,2	108,4	0,0	0,0	0,0
2000	23,3	32,6	44	314,7	1181,2	123,1	0,0	0,0	0,0
2100	22,2	32,4	44	314,7	1130,1	135,3	0,0	0,0	0,0
2200	21,2	32,2	45	314,7	1082,2	146,1	0,0	0,0	0,0
2300	20,4	32,0	45	314,7	1040,8	154,4	0,0	0,0	0,0

**Air System Information**

Air System Name .....	Sistema Escritório	Number of zones .....	1
Equipment Class .....	TERM	Floor Area .....	25,0 m²
Air System Type .....	PKG-FC	Location .....	Lisbon, Portugal

**Sizing Calculation Information**

Calculation Months .....	Jan to Dec	Zone L/s Sizing .....	Sum of space airflow rates
Sizing Data .....	Calculated	Space L/s Sizing .....	Individual peak space loads

## Air System Information

Air System Name ..... **Sistema Escritório**  
Equipment Class ..... **TERM**  
Air System Type ..... **PKG-FC**

Number of zones ..... **1**  
Floor Area ..... **25,0** m²  
Location ..... **Lisbon, Portugal**

## Sizing Calculation Information

Calculation Months ..... **Jan to Dec**  
Sizing Data ..... **Calculated**

Zone L/s Sizing ..... **Sum of space airflow rates**  
Space L/s Sizing ..... **Individual peak space loads**

## Zone Sizing Data

Zone Name	Maximum Cooling Sensible (kW)	Design Airflow (L/s)	Minimum Airflow (L/s)	Time of Peak Load	Maximum Heating Load (kW)	Zone Floor Area (m²)	Zone L/(s-m²)
Zone 1	2,7	269	269	Jul 1400	1,2	25,0	10,75

## Terminal Unit Sizing Data - Cooling

Zone Name	Total Coil Load (kW)	Sens Coil Load (kW)	Coil Entering DB / WB (°C)	Coil Leaving DB / WB (°C)	Water Flow @ 5,6 °K (L/s)	Time of Peak Load
Zone 1	0,0	0,0	-18,3 / -18,3	-18,3 / -18,3	0,00	Des 0000

## Terminal Unit Sizing Data - Heating, Fan, Ventilation

Zone Name	Heating Coil Load (kW)	Heating Coil Ent/Lvg DB (°C)	Htg Coil Water Flow @11,1 °K (L/s)	Fan Design Airflow (L/s)	Fan Motor (BHP)	Fan Motor (kW)	OA Vent Design Airflow (L/s)
Zone 1	0,0	-18,3 / -18,3	0,00	269	0,000	0,000	12

## Space Loads and Airflows

Zone Name / Space Name	Mult.	Cooling Sensible (kW)	Time of Load	Air Flow (L/s)	Heating Load (kW)	Floor Area (m²)	Space L/(s-m²)
<b>Zone 1</b>							
Escritório	1	2,7	Jul 1400	269	1,2	25,0	10,75

	DESIGN COOLING			DESIGN HEATING		
	NO COOLING DATA NO COOLING OA DB / WB			HEATING DATA AT DES HTG HEATING OA DB / WB 5,0 °C / 4,3 °C		
ZONE LOADS	Details	Sensible (W)	Latent (W)	Details	Sensible (W)	Latent (W)
Window & Skylight Solar Loads	8 m²	-	-	8 m²	-	-
Wall Transmission	22 m²	-	-	22 m²	161	-
Roof Transmission	25 m²	-	-	25 m²	175	-
Window Transmission	8 m²	-	-	8 m²	517	-
Skylight Transmission	0 m²	-	-	0 m²	0	-
Door Loads	0 m²	-	-	0 m²	0	-
Floor Transmission	25 m²	-	-	25 m²	344	-
Partitions	28 m²	-	-	28 m²	0	-
Ceiling	0 m²	-	-	0 m²	0	-
Overhead Lighting	-	-	-	0	0	-
Task Lighting	-	-	-	0	0	-
Electric Equipment	-	-	-	0	0	-
People	-	-	-	0	0	0
Infiltration	-	-	-	-	0	0
Miscellaneous	-	-	-	-	0	0
Safety Factor	0% / 0%	-	-	0%	0	0
>> Total Zone Loads	-	-	-	-	1196	0
Zone Conditioning	-	-	-	-	-70	0
Plenum Wall Load	0%	-	-	0	0	-
Plenum Roof Load	0%	-	-	0	0	-
Plenum Lighting Load	0%	-	-	0	0	-
Exhaust Fan Load	-	-	-	0 L/s	0	-
Ventilation Load	-	-	-	12 L/s	74	0
Ventilation Fan Load	-	-	-	0 L/s	0	-
Space Fan Coil Fans	-	-	-	-	0	-
Duct Heat Gain / Loss	0%	-	-	0%	0	-
>> Total System Loads	-	-	-	-	4	0
Terminal Unit Cooling	-	-	-	-	0	0
Terminal Unit Heating	-	-	-	-	0	-
>> Total Conditioning	-	-	-	-	0	0
Key:	Positive values are clg loads Negative values are htg loads			Positive values are htg loads Negative values are clg loads		

Zone 1	DESIGN COOLING			DESIGN HEATING		
	COOLING DATA AT Jul 1400			HEATING DATA AT DES HTG		
	COOLING OA DB / WB 32,7 °C / 22,2 °C			HEATING OA DB / WB 5,0 °C / 4,3 °C		
	OCCUPIED T-STAT 23,0 °C			OCCUPIED T-STAT 22,0 °C		
ZONE LOADS	Details	Sensible (W)	Latent (W)	Details	Sensible (W)	Latent (W)
Window & Skylight Solar Loads	8 m²	1390	-	8 m²	-	-
Wall Transmission	22 m²	65	-	22 m²	161	-
Roof Transmission	25 m²	74	-	25 m²	175	-
Window Transmission	8 m²	240	-	8 m²	517	-
Skylight Transmission	0 m²	0	-	0 m²	0	-
Door Loads	0 m²	0	-	0 m²	0	-
Floor Transmission	25 m²	160	-	25 m²	344	-
Partitions	28 m²	48	-	28 m²	0	-
Ceiling	0 m²	0	-	0 m²	0	-
Overhead Lighting	263 W	262	-	0	0	-
Task Lighting	0 W	0	-	0	0	-
Electric Equipment	375 W	375	-	0	0	-
People	2	125	92	0	0	0
Infiltration	-	0	0	-	0	0
Miscellaneous	-	0	0	-	0	0
Safety Factor	0% / 0%	0	0	0%	0	0
>> Total Zone Loads	-	2739	92	-	1196	0

TABLE 1.1.A. COMPONENT LOADS FOR SPACE "Escritório" IN ZONE "Zone 1"						
	DESIGN COOLING			DESIGN HEATING		
	COOLING DATA AT Jul 1400 COOLING OA DB / WB 32,7 °C / 22,2 °C OCCUPIED T-STAT 23,0 °C			HEATING DATA AT DES HTG HEATING OA DB / WB 5,0 °C / 4,3 °C OCCUPIED T-STAT 22,0 °C		
		Sensible	Latent		Sensible	Latent
SPACE LOADS	Details	(W)	(W)	Details	(W)	(W)
Window & Skylight Solar Loads	8 m²	1390	-	8 m²	-	-
Wall Transmission	22 m²	65	-	22 m²	161	-
Roof Transmission	25 m²	74	-	25 m²	175	-
Window Transmission	8 m²	240	-	8 m²	517	-
Skylight Transmission	0 m²	0	-	0 m²	0	-
Door Loads	0 m²	0	-	0 m²	0	-
Floor Transmission	25 m²	160	-	25 m²	344	-
Partitions	28 m²	48	-	28 m²	0	-
Ceiling	0 m²	0	-	0 m²	0	-
Overhead Lighting	263 W	262	-	0	0	-
Task Lighting	0 W	0	-	0	0	-
Electric Equipment	375 W	375	-	0	0	-
People	2	125	92	0	0	0
Infiltration	-	0	0	-	0	0
Miscellaneous	-	0	0	-	0	0
Safety Factor	0% / 0%	0	0	0%	0	0
>> Total Zone Loads	-	2739	92	-	1196	0

TABLE 1.1.B. ENVELOPE LOADS FOR SPACE "Escritório" IN ZONE "Zone 1"						
				COOLING	COOLING	HEATING
	Area	U-Value	Shade	TRANS	SOLAR	TRANS
	(m²)	(W/(m²·°K))	Coeff.	(W)	(W)	(W)
<b>N EXPOSURE</b>						
WALL	11	0,603	-	39	-	113
WINDOW 1	4	5,100	1,000	161	561	347
<b>E EXPOSURE</b>						
WALL	11	0,257	-	25	-	48
WINDOW 1	4	2,500	1,000	79	829	170
<b>H EXPOSURE</b>						
ROOF	25	0,411	-	74	-	175



ZONE: Zone 1 DESIGN MONTH: JANUARY									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	19,2	32,1	42	268,7	887,2	175,2	0,0	0,0	0,0
0100	18,7	31,9	43	268,7	854,5	177,8	0,0	0,0	0,0
0200	18,2	31,7	43	268,7	821,4	181,9	0,0	0,0	0,0
0300	17,8	31,5	44	268,7	791,4	184,7	0,0	0,0	0,0
0400	17,5	31,3	44	268,7	764,5	186,3	0,0	0,0	0,0
0500	17,4	31,2	45	268,7	744,3	185,6	0,0	0,0	0,0
0600	17,6	31,0	45	268,7	734,6	181,5	0,0	0,0	0,0
0700	18,1	30,9	45	268,7	735,6	173,9	0,0	0,0	0,0
0800	19,0	31,2	45	268,7	853,7	164,9	0,0	0,0	0,0
0900	20,3	31,7	43	268,7	1053,4	162,4	0,0	0,0	0,0
1000	21,8	32,1	42	268,7	1176,1	135,5	0,0	0,0	0,0
1100	23,5	32,4	42	268,7	1250,4	115,9	0,0	0,0	0,0
1200	25,1	32,7	41	268,7	1294,7	96,4	0,0	0,0	0,0
1300	26,3	32,9	41	268,7	1352,4	85,2	0,0	0,0	0,0
1400	27,1	33,1	40	268,7	1382,5	77,3	0,0	0,0	0,0
1500	27,4	33,2	40	268,7	1373,8	74,0	0,0	0,0	0,0
1600	27,1	33,1	40	268,7	1320,5	78,4	0,0	0,0	0,0
1700	26,4	33,0	41	268,7	1238,7	83,5	0,0	0,0	0,0
1800	25,3	32,8	41	268,7	1184,6	96,5	0,0	0,0	0,0
1900	24,0	32,6	41	268,7	1125,0	115,5	0,0	0,0	0,0
2000	22,7	32,4	42	268,7	1066,0	130,3	0,0	0,0	0,0
2100	21,6	32,2	42	268,7	1013,9	142,6	0,0	0,0	0,0
2200	20,6	32,0	43	268,7	965,2	153,4	0,0	0,0	0,0
2300	19,8	31,8	43	268,7	922,9	161,7	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: FEBRUARY									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	20,3	33,5	40	268,7	1018,4	178,8	0,0	0,0	0,0
0100	19,8	33,3	40	268,7	982,8	181,1	0,0	0,0	0,0
0200	19,3	33,0	41	268,7	947,0	185,1	0,0	0,0	0,0
0300	18,9	32,8	41	268,7	914,3	187,7	0,0	0,0	0,0
0400	18,6	32,6	42	268,7	884,8	189,1	0,0	0,0	0,0
0500	18,5	32,5	42	268,7	862,1	188,2	0,0	0,0	0,0
0600	18,7	32,3	43	268,7	850,0	183,9	0,0	0,0	0,0
0700	19,2	32,2	43	268,7	848,8	176,1	0,0	0,0	0,0
0800	20,1	32,7	42	268,7	1063,1	170,7	0,0	0,0	0,0
0900	21,4	33,2	41	268,7	1220,7	157,0	0,0	0,0	0,0
1000	22,9	33,5	40	268,7	1333,2	139,2	0,0	0,0	0,0
1100	24,6	33,8	39	268,7	1406,9	124,0	0,0	0,0	0,0
1200	26,2	34,1	39	268,7	1451,3	105,3	0,0	0,0	0,0
1300	27,4	34,4	38	268,7	1508,9	89,3	0,0	0,0	0,0
1400	28,2	34,6	38	268,7	1542,5	81,5	0,0	0,0	0,0
1500	28,5	34,7	38	268,7	1540,2	78,9	0,0	0,0	0,0
1600	28,2	34,6	38	268,7	1496,3	83,7	0,0	0,0	0,0
1700	27,5	34,5	38	268,7	1417,2	88,8	0,0	0,0	0,0
1800	26,4	34,2	39	268,7	1336,8	104,6	0,0	0,0	0,0
1900	25,1	34,0	39	268,7	1273,0	119,6	0,0	0,0	0,0
2000	23,8	33,8	40	268,7	1210,2	134,2	0,0	0,0	0,0
2100	22,7	33,6	40	268,7	1154,6	146,2	0,0	0,0	0,0
2200	21,7	33,4	41	268,7	1102,5	156,9	0,0	0,0	0,0
2300	20,9	33,2	41	268,7	1057,1	165,1	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: MARCH									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	22,0	36,3	36	268,7	1282,6	194,7	0,0	0,0	0,0
0100	21,5	36,1	37	268,7	1238,9	196,3	0,0	0,0	0,0
0200	21,0	35,8	37	268,7	1195,2	199,7	0,0	0,0	0,0
0300	20,6	35,6	38	268,7	1154,9	201,8	0,0	0,0	0,0
0400	20,3	35,3	38	268,7	1118,2	202,6	0,0	0,0	0,0
0500	20,2	35,1	39	268,7	1088,6	201,2	0,0	0,0	0,0
0600	20,4	34,9	39	268,7	1069,9	196,3	0,0	0,0	0,0
0700	20,9	35,2	39	268,7	1236,2	194,8	0,0	0,0	0,0
0800	21,8	35,8	38	268,7	1440,6	190,2	0,0	0,0	0,0
0900	23,1	36,2	37	268,7	1594,3	177,6	0,0	0,0	0,0
1000	24,6	36,6	36	268,7	1708,1	161,6	0,0	0,0	0,0
1100	26,3	36,9	35	268,7	1779,3	141,7	0,0	0,0	0,0
1200	27,9	37,1	35	268,7	1823,7	122,9	0,0	0,0	0,0
1300	29,1	37,4	34	268,7	1886,4	109,8	0,0	0,0	0,0
1400	29,9	37,7	34	268,7	1920,8	103,5	0,0	0,0	0,0
1500	30,2	37,8	34	268,7	1916,3	97,2	0,0	0,0	0,0
1600	29,9	37,7	34	268,7	1867,4	102,7	0,0	0,0	0,0
1700	29,2	37,5	34	268,7	1780,5	112,2	0,0	0,0	0,0
1800	28,1	37,2	35	268,7	1661,7	122,1	0,0	0,0	0,0
1900	26,8	37,0	35	268,7	1584,6	136,9	0,0	0,0	0,0
2000	25,5	36,7	36	268,7	1511,1	151,0	0,0	0,0	0,0
2100	24,4	36,5	36	268,7	1445,7	162,7	0,0	0,0	0,0
2200	23,4	36,2	37	268,7	1384,2	172,8	0,0	0,0	0,0
2300	22,6	36,0	37	268,7	1330,0	180,5	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: APRIL									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	22,5	39,8	31	268,7	1592,0	234,4	0,0	0,0	0,0
0100	22,0	39,5	32	268,7	1533,2	234,5	0,0	0,0	0,0
0200	21,5	39,2	32	268,7	1475,1	236,9	0,0	0,0	0,0
0300	21,1	38,8	33	268,7	1421,0	238,0	0,0	0,0	0,0
0400	20,8	38,5	33	268,7	1371,0	237,8	0,0	0,0	0,0
0500	20,7	38,2	34	268,7	1328,8	235,3	0,0	0,0	0,0
0600	20,9	38,3	34	268,7	1423,7	236,3	0,0	0,0	0,0
0700	21,4	38,9	33	268,7	1697,7	237,0	0,0	0,0	0,0
0800	22,3	39,4	32	268,7	1907,8	230,2	0,0	0,0	0,0
0900	23,6	39,9	31	268,7	2077,9	218,9	0,0	0,0	0,0
1000	25,1	40,4	30	268,7	2200,7	207,7	0,0	0,0	0,0
1100	26,8	40,7	30	268,7	2270,4	188,1	0,0	0,0	0,0
1200	28,4	41,0	29	268,7	2321,3	170,1	0,0	0,0	0,0
1300	29,6	41,3	29	268,7	2391,9	156,0	0,0	0,0	0,0
1400	30,4	41,6	28	268,7	2427,6	150,6	0,0	0,0	0,0
1500	30,7	41,7	28	268,7	2417,8	146,1	0,0	0,0	0,0
1600	30,4	41,6	28	268,7	2358,3	146,7	0,0	0,0	0,0
1700	29,7	41,4	29	268,7	2254,8	156,9	0,0	0,0	0,0
1800	28,6	41,0	29	268,7	2114,5	166,0	0,0	0,0	0,0
1900	27,3	40,7	30	268,7	1980,7	178,1	0,0	0,0	0,0
2000	26,0	40,3	30	268,7	1887,8	191,9	0,0	0,0	0,0
2100	24,9	40,0	31	268,7	1804,1	202,7	0,0	0,0	0,0
2200	23,9	39,7	32	268,7	1725,6	211,9	0,0	0,0	0,0
2300	23,1	39,4	32	268,7	1655,0	218,7	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: MAY									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	23,1	40,3	31	268,7	1640,9	232,6	0,0	0,0	0,0
0100	22,6	40,0	32	268,7	1581,8	232,8	0,0	0,0	0,0
0200	22,1	39,6	32	268,7	1523,3	235,2	0,0	0,0	0,0
0300	21,7	39,3	33	268,7	1468,9	236,2	0,0	0,0	0,0
0400	21,4	39,0	34	268,7	1418,6	236,0	0,0	0,0	0,0
0500	21,3	38,7	34	268,7	1376,0	233,5	0,0	0,0	0,0
0600	21,5	39,1	33	268,7	1602,8	238,7	0,0	0,0	0,0
0700	22,0	39,5	33	268,7	1797,2	238,1	0,0	0,0	0,0
0800	22,9	40,0	32	268,7	1961,6	231,7	0,0	0,0	0,0
0900	24,2	40,4	31	268,7	2108,8	220,4	0,0	0,0	0,0
1000	25,7	40,8	31	268,7	2217,1	204,8	0,0	0,0	0,0
1100	27,4	41,1	30	268,7	2284,9	185,3	0,0	0,0	0,0
1200	29,0	41,4	30	268,7	2341,2	167,6	0,0	0,0	0,0
1300	30,2	41,7	29	268,7	2408,6	152,9	0,0	0,0	0,0
1400	31,0	41,9	29	268,7	2444,7	147,7	0,0	0,0	0,0
1500	31,3	42,1	29	268,7	2441,3	143,6	0,0	0,0	0,0
1600	31,0	42,0	29	268,7	2392,2	144,7	0,0	0,0	0,0
1700	30,3	41,9	29	268,7	2313,2	155,8	0,0	0,0	0,0
1800	29,2	41,6	29	268,7	2207,0	166,4	0,0	0,0	0,0
1900	27,9	41,1	30	268,7	2033,5	176,0	0,0	0,0	0,0
2000	26,6	40,8	31	268,7	1938,9	190,3	0,0	0,0	0,0
2100	25,5	40,5	31	268,7	1854,5	201,0	0,0	0,0	0,0
2200	24,5	40,2	32	268,7	1775,3	210,3	0,0	0,0	0,0
2300	23,7	39,9	32	268,7	1704,3	217,0	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: JUNE									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	24,2	41,5	30	268,7	1759,4	233,3	0,0	0,0	0,0
0100	23,7	41,1	31	268,7	1698,1	233,3	0,0	0,0	0,0
0200	23,2	40,8	31	268,7	1637,5	235,6	0,0	0,0	0,0
0300	22,8	40,4	32	268,7	1581,0	236,6	0,0	0,0	0,0
0400	22,5	40,1	33	268,7	1528,8	236,3	0,0	0,0	0,0
0500	22,4	39,8	33	268,7	1488,6	233,9	0,0	0,0	0,0
0600	22,6	40,3	32	268,7	1733,8	240,4	0,0	0,0	0,0
0700	23,1	40,7	32	268,7	1921,9	238,7	0,0	0,0	0,0
0800	24,0	41,1	31	268,7	2075,8	231,6	0,0	0,0	0,0
0900	25,3	41,5	30	268,7	2221,1	220,6	0,0	0,0	0,0
1000	26,8	41,9	30	268,7	2331,9	205,2	0,0	0,0	0,0
1100	28,5	42,2	29	268,7	2402,7	185,8	0,0	0,0	0,0
1200	30,1	42,5	29	268,7	2457,7	168,0	0,0	0,0	0,0
1300	31,3	42,9	28	268,7	2525,2	153,2	0,0	0,0	0,0
1400	32,1	43,1	28	268,7	2564,7	143,7	0,0	0,0	0,0
1500	32,4	43,2	28	268,7	2564,6	144,2	0,0	0,0	0,0
1600	32,1	43,2	28	268,7	2520,2	145,7	0,0	0,0	0,0
1700	31,4	43,1	28	268,7	2453,2	157,3	0,0	0,0	0,0
1800	30,3	42,9	28	268,7	2358,7	168,5	0,0	0,0	0,0
1900	29,0	42,4	29	268,7	2187,7	178,1	0,0	0,0	0,0
2000	27,7	42,0	30	268,7	2068,0	191,0	0,0	0,0	0,0
2100	26,6	41,7	30	268,7	1980,6	202,0	0,0	0,0	0,0
2200	25,6	41,4	31	268,7	1898,7	211,2	0,0	0,0	0,0
2300	24,8	41,1	31	268,7	1825,1	217,8	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: JULY									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	24,8	42,6	29	268,7	1867,4	241,5	0,0	0,0	0,0
0100	24,3	42,3	29	268,7	1802,3	241,2	0,0	0,0	0,0
0200	23,8	41,9	30	268,7	1738,0	243,2	0,0	0,0	0,0
0300	23,4	41,5	30	268,7	1678,0	243,9	0,0	0,0	0,0
0400	23,1	41,2	31	268,7	1622,4	243,4	0,0	0,0	0,0
0500	23,0	40,9	31	268,7	1574,8	240,5	0,0	0,0	0,0
0600	23,2	41,2	31	268,7	1784,6	244,0	0,0	0,0	0,0
0700	23,7	41,8	30	268,7	2016,7	246,6	0,0	0,0	0,0
0800	24,6	42,2	29	268,7	2201,6	240,4	0,0	0,0	0,0
0900	25,9	42,7	28	268,7	2370,0	225,3	0,0	0,0	0,0
1000	27,4	43,2	28	268,7	2495,0	214,7	0,0	0,0	0,0
1100	29,1	43,5	27	268,7	2573,5	195,6	0,0	0,0	0,0
1200	30,7	43,8	27	268,7	2626,3	177,5	0,0	0,0	0,0
1300	31,9	44,1	26	268,7	2698,6	163,1	0,0	0,0	0,0
1400	32,7	44,4	26	268,7	2739,2	158,4	0,0	0,0	0,0
1500	33,0	44,5	26	268,7	2738,7	154,5	0,0	0,0	0,0
1600	32,7	44,5	26	268,7	2689,6	155,6	0,0	0,0	0,0
1700	32,0	44,4	26	268,7	2605,7	166,8	0,0	0,0	0,0
1800	30,9	44,1	27	268,7	2496,0	177,4	0,0	0,0	0,0
1900	29,6	43,6	27	268,7	2307,6	186,3	0,0	0,0	0,0
2000	28,3	43,2	28	268,7	2192,9	199,7	0,0	0,0	0,0
2100	27,2	42,9	28	268,7	2100,9	210,4	0,0	0,0	0,0
2200	26,2	42,5	29	268,7	2014,7	219,4	0,0	0,0	0,0
2300	25,4	42,2	29	268,7	1937,1	225,8	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: AUGUST									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	24,8	41,6	30	268,7	1774,8	227,7	0,0	0,0	0,0
0100	24,3	41,3	31	268,7	1714,7	227,9	0,0	0,0	0,0
0200	23,8	40,9	31	268,7	1655,3	230,2	0,0	0,0	0,0
0300	23,4	40,6	32	268,7	1600,0	231,2	0,0	0,0	0,0
0400	23,1	40,3	32	268,7	1548,9	231,0	0,0	0,0	0,0
0500	23,0	40,0	33	268,7	1505,6	228,5	0,0	0,0	0,0
0600	23,2	40,0	33	268,7	1573,6	228,0	0,0	0,0	0,0
0700	23,7	40,6	32	268,7	1854,8	230,4	0,0	0,0	0,0
0800	24,6	41,2	31	268,7	2074,0	224,3	0,0	0,0	0,0
0900	25,9	41,7	30	268,7	2252,2	213,2	0,0	0,0	0,0
1000	27,4	42,1	29	268,7	2383,1	197,6	0,0	0,0	0,0
1100	29,1	42,5	29	268,7	2460,1	181,9	0,0	0,0	0,0
1200	30,7	42,8	28	268,7	2511,8	159,3	0,0	0,0	0,0
1300	31,9	43,1	28	268,7	2584,2	150,2	0,0	0,0	0,0
1400	32,7	43,4	27	268,7	2623,4	140,7	0,0	0,0	0,0
1500	33,0	43,5	27	268,7	2616,6	140,4	0,0	0,0	0,0
1600	32,7	43,5	27	268,7	2559,5	141,3	0,0	0,0	0,0
1700	32,0	43,3	28	268,7	2457,1	151,3	0,0	0,0	0,0
1800	30,9	42,9	28	268,7	2315,3	160,4	0,0	0,0	0,0
1900	29,6	42,5	29	268,7	2171,4	172,0	0,0	0,0	0,0
2000	28,3	42,2	29	268,7	2076,7	185,9	0,0	0,0	0,0
2100	27,2	41,8	30	268,7	1991,4	196,6	0,0	0,0	0,0
2200	26,2	41,5	30	268,7	1911,2	205,8	0,0	0,0	0,0
2300	25,4	41,2	31	268,7	1839,2	212,5	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: SEPTEMBER									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	23,7	38,6	35	268,7	1495,0	202,1	0,0	0,0	0,0
0100	23,2	38,3	36	268,7	1445,9	203,2	0,0	0,0	0,0
0200	22,7	38,0	36	268,7	1397,1	206,3	0,0	0,0	0,0
0300	22,3	37,8	37	268,7	1352,0	208,0	0,0	0,0	0,0
0400	22,0	37,5	37	268,7	1310,7	208,5	0,0	0,0	0,0
0500	21,9	37,2	38	268,7	1276,7	206,7	0,0	0,0	0,0
0600	22,1	37,0	38	268,7	1253,8	201,5	0,0	0,0	0,0
0700	22,6	37,6	37	268,7	1505,9	203,8	0,0	0,0	0,0
0800	23,5	38,1	36	268,7	1728,1	197,2	0,0	0,0	0,0
0900	24,8	38,6	35	268,7	1895,7	188,0	0,0	0,0	0,0
1000	26,3	39,0	34	268,7	2011,6	171,5	0,0	0,0	0,0
1100	28,0	39,3	34	268,7	2075,3	150,5	0,0	0,0	0,0
1200	29,6	39,6	33	268,7	2129,9	133,7	0,0	0,0	0,0
1300	30,8	40,0	33	268,7	2195,8	119,9	0,0	0,0	0,0
1400	31,6	40,2	32	268,7	2225,5	113,3	0,0	0,0	0,0
1500	31,9	40,3	32	268,7	2209,9	111,6	0,0	0,0	0,0
1600	31,6	40,2	32	268,7	2143,7	111,9	0,0	0,0	0,0
1700	30,9	39,9	32	268,7	2033,2	121,0	0,0	0,0	0,0
1800	29,8	39,6	33	268,7	1908,6	130,9	0,0	0,0	0,0
1900	28,5	39,3	34	268,7	1826,6	145,5	0,0	0,0	0,0
2000	27,2	39,1	34	268,7	1746,8	159,3	0,0	0,0	0,0
2100	26,1	38,8	34	268,7	1675,1	170,6	0,0	0,0	0,0
2200	25,1	38,5	35	268,7	1607,8	180,4	0,0	0,0	0,0
2300	24,3	38,3	35	268,7	1547,8	187,8	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: OCTOBER									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	22,6	36,3	39	268,7	1282,9	186,5	0,0	0,0	0,0
0100	22,1	36,1	40	268,7	1241,2	188,3	0,0	0,0	0,0
0200	21,6	35,8	40	268,7	1199,5	191,8	0,0	0,0	0,0
0300	21,2	35,6	41	268,7	1161,2	194,0	0,0	0,0	0,0
0400	20,9	35,4	41	268,7	1126,4	195,0	0,0	0,0	0,0
0500	20,8	35,1	42	268,7	1098,6	193,7	0,0	0,0	0,0
0600	21,0	35,0	42	268,7	1081,7	188,9	0,0	0,0	0,0
0700	21,5	35,2	41	268,7	1211,7	184,9	0,0	0,0	0,0
0800	22,4	35,8	40	268,7	1458,1	188,7	0,0	0,0	0,0
0900	23,7	36,3	39	268,7	1621,7	170,4	0,0	0,0	0,0
1000	25,2	36,7	38	268,7	1727,5	154,9	0,0	0,0	0,0
1100	26,9	36,9	37	268,7	1782,9	133,7	0,0	0,0	0,0
1200	28,5	37,2	37	268,7	1841,4	113,4	0,0	0,0	0,0
1300	29,7	37,5	36	268,7	1900,7	103,7	0,0	0,0	0,0
1400	30,5	37,7	36	268,7	1922,6	95,9	0,0	0,0	0,0
1500	30,8	37,8	36	268,7	1899,1	93,1	0,0	0,0	0,0
1600	30,5	37,7	36	268,7	1825,0	92,9	0,0	0,0	0,0
1700	29,8	37,4	36	268,7	1713,1	101,6	0,0	0,0	0,0
1800	28,7	37,2	37	268,7	1643,4	114,4	0,0	0,0	0,0
1900	27,4	37,0	37	268,7	1571,9	128,7	0,0	0,0	0,0
2000	26,1	36,7	38	268,7	1501,7	142,9	0,0	0,0	0,0
2100	25,0	36,5	38	268,7	1439,0	154,6	0,0	0,0	0,0
2200	24,0	36,2	39	268,7	1380,1	164,8	0,0	0,0	0,0
2300	23,2	36,0	39	268,7	1328,1	172,6	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: NOVEMBER									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	20,9	33,7	45	268,7	1035,4	173,7	0,0	0,0	0,0
0100	20,4	33,5	45	268,7	1000,8	176,2	0,0	0,0	0,0
0200	19,9	33,2	46	268,7	966,0	180,2	0,0	0,0	0,0
0300	19,5	33,0	46	268,7	934,3	182,9	0,0	0,0	0,0
0400	19,2	32,8	47	268,7	905,8	184,3	0,0	0,0	0,0
0500	19,1	32,7	47	268,7	884,1	183,5	0,0	0,0	0,0
0600	19,3	32,5	47	268,7	872,9	179,2	0,0	0,0	0,0
0700	19,8	32,4	48	268,7	872,6	171,5	0,0	0,0	0,0
0800	20,7	33,0	46	268,7	1094,8	167,0	0,0	0,0	0,0
0900	22,0	33,4	45	268,7	1270,1	156,7	0,0	0,0	0,0
1000	23,5	33,8	44	268,7	1377,0	136,5	0,0	0,0	0,0
1100	25,2	34,1	43	268,7	1434,6	119,0	0,0	0,0	0,0
1200	26,8	34,4	43	268,7	1490,3	99,8	0,0	0,0	0,0
1300	28,0	34,6	42	268,7	1546,5	87,2	0,0	0,0	0,0
1400	28,8	34,8	42	268,7	1566,8	76,2	0,0	0,0	0,0
1500	29,1	34,9	41	268,7	1543,3	72,3	0,0	0,0	0,0
1600	28,8	34,7	42	268,7	1471,5	76,5	0,0	0,0	0,0
1700	28,1	34,6	42	268,7	1401,4	83,2	0,0	0,0	0,0
1800	27,0	34,4	42	268,7	1345,3	95,6	0,0	0,0	0,0
1900	25,7	34,2	43	268,7	1283,5	114,6	0,0	0,0	0,0
2000	24,4	34,0	43	268,7	1222,4	129,2	0,0	0,0	0,0
2100	23,3	33,8	44	268,7	1168,2	141,3	0,0	0,0	0,0
2200	22,3	33,6	44	268,7	1117,4	152,0	0,0	0,0	0,0
2300	21,5	33,4	45	268,7	1073,1	160,2	0,0	0,0	0,0

ZONE: Zone 1 DESIGN MONTH: DECEMBER									
Hour	OA TEMP (°C)	ZONE TEMP (°C)	RH (%)	ZONE AIRFLOW (L/s)	ZONE SENSIBLE LOAD (W)	ZONE COND (W)	TERMINAL COOLING COIL (W)	TERMINAL HEATING COIL (W)	ZONE HEATING UNIT (W)
0000	19,8	32,4	45	268,7	911,6	171,2	0,0	0,0	0,0
0100	19,3	32,2	45	268,7	879,7	173,8	0,0	0,0	0,0
0200	18,8	32,0	46	268,7	847,3	177,9	0,0	0,0	0,0
0300	18,4	31,8	46	268,7	818,0	180,8	0,0	0,0	0,0
0400	18,1	31,6	47	268,7	791,8	182,5	0,0	0,0	0,0
0500	18,0	31,4	47	268,7	772,3	181,8	0,0	0,0	0,0
0600	18,2	31,3	47	268,7	763,2	177,7	0,0	0,0	0,0
0700	18,7	31,2	48	268,7	764,8	170,1	0,0	0,0	0,0
0800	19,6	31,5	47	268,7	877,3	160,9	0,0	0,0	0,0
0900	20,9	32,0	46	268,7	1085,1	159,1	0,0	0,0	0,0
1000	22,4	32,4	44	268,7	1205,2	133,2	0,0	0,0	0,0
1100	24,1	32,7	44	268,7	1271,8	111,6	0,0	0,0	0,0
1200	25,7	32,9	43	268,7	1319,5	93,9	0,0	0,0	0,0
1300	26,9	33,2	42	268,7	1377,1	83,9	0,0	0,0	0,0
1400	27,7	33,4	42	268,7	1401,2	73,4	0,0	0,0	0,0
1500	28,0	33,4	42	268,7	1383,0	69,1	0,0	0,0	0,0
1600	27,7	33,3	42	268,7	1318,0	73,0	0,0	0,0	0,0
1700	27,0	33,2	42	268,7	1255,7	80,0	0,0	0,0	0,0
1800	25,9	33,1	43	268,7	1203,3	92,4	0,0	0,0	0,0
1900	24,6	32,9	43	268,7	1145,0	111,3	0,0	0,0	0,0
2000	23,3	32,6	44	268,7	1087,0	126,1	0,0	0,0	0,0
2100	22,2	32,5	44	268,7	1035,9	138,4	0,0	0,0	0,0
2200	21,2	32,2	45	268,7	988,0	149,2	0,0	0,0	0,0
2300	20,4	32,1	45	268,7	946,6	157,6	0,0	0,0	0,0

**Anexo F – Comparação entre os valores do  
HAP e da aplicação nas cargas térmicas  
horárias**





HAP	1 de Julho	15 de Julho	31 de Julho
1961,600	-102,9377648	30,81320238	16,7341532
1896,400	-159,2539616	-18,46346975	-32,54251893
1832,200	-208,5306337	-53,6610927	-74,77966647
1772,200	-250,7677812	-88,85871565	-109,9772894
1716,600	-278,9258796	-117,016814	-138,1353878
1669,000	-300,0444534	-131,0958632	-152,214437
1878,800	-307,0839779	-131,0958632	-77,37896155
2110,900	-271,886355	-102,9377648	146,5686614
2295,800	-166,2934861	-25,50299434	403,5429565
2464,200	-18,46346975	94,16892369	715,7148745
2589,200	143,4455958	227,9198909	914,0053663
2667,700	284,2360876	340,5522843	1014,927284
2720,400	389,8289565	425,0265794	818,941104
2792,800	425,0265794	453,1846778	924,5992024
2833,300	417,9870548	446,1451532	1088,809678
2832,900	403,9080057	432,066104	1132,230629
2783,800	368,7103827	410,9475302	989,0725302
2699,900	326,4732352	375,7499073	762,6249073
2590,200	277,196563	340,5522843	510,3877598
2401,800	220,8803663	291,2756122	354,4006122
2287,000	157,524645	241,9989401	234,9594155
2195,100	94,16892369	185,6827434	178,6432188
2108,900	23,77367779	136,4060712	122,3270221
2031,300	-39,58204352	80,08987451	66,01082533



## **Anexo G – Relatório Análise de Projecto**



	A	B	C	D	E	F
1	Carga_envolvente_externa_W_1	Carga_envolvente_externa_W_2	Carga_envolvente_externa_W_3	Carga_envolvente_externa_W_4	Carga_envolvente_externa_W_5	Carga_envolvente_externa_W_6
2	72,963	31,097	113,025	222,75	224,4	110

	G	H	I	J	K	L
1	Carga_envolvente_interna_W_1	Carga_envolvente_interna_W_2	Carga_envolvente_interna_W_3	Carga_ocupacao_sensivel_W	Carga_ocupacao_latente_W	Carga_diversa_sensivel_W
2	56,688795	15,84	15,62085	125	91,66666667	0

	M	N	O	P	Q	R	S
1	Carga_diversa_latente_W	Carga_iluminacao_W	Carga_equipamentos_W	Carga_infiltracoes_sensivel_W	Carga_infiltracoes_total_W	Carga_infiltracoes_latente_W	Carga_sensivel_de_arrefecimento_W
2	0	262,5	375	0,1127049	0,226104949	0,113400049	1624,99735

	T	U	V	W	X
1	Carga_latente_de_arrefecimento_W	Carga_total_de_arrefecimento_W	Carga_envolvente_externa_inverno_W_1	Carga_envolvente_externa_inverno_W_2	Carga_envolvente_externa_inverno_W_3
2	91,7800672	1716,777417	-112,761	-48,059	-174,675

	Y	Z	AA	AB	AC
1	Carga_envolvente_externa_inverno_W_4	Carga_envolvente_externa_inverno_W_5	Carga_envolvente_externa_inverno_W_6	Carga_infiltracoes_sensivel_inverno_W	Carga_infiltracoes_total_inverno_W
2	-344,25	-346,8	-170	-0,1741803	-0,259245003

	AD	AE	AF	AG
1	Carga_infiltracoes_latente_inverno_W	Carga_sensivel_de_aquecimento_W	Carga_latente_de_aquecimento_W	Carga_total_de_aquecimento_W
2	-0,085064703	-1196,71918	-0,085064703	-1196,804245



## **Anexo H – Relatórios Análise Anual**





▲	A	B	C	D	E	F	G	H	I
1	Dia	Hora	Dia_da_semana_ou_Feriado	Horario_de_Ocupacao	Horario_de_Iluminacao	Horario_de_equipamentos	Horario_de_cargas_diversas	Carga_ocupacao_sensivel_W	Carga_ocupacao_latente_W
2	1	1	Feriado	0	5	15	0	0	0
3	1	2	Feriado	0	5	15	0	0	0
4	1	3	Feriado	0	5	15	0	0	0
5	1	4	Feriado	0	5	15	0	0	0
6	1	5	Feriado	0	5	15	0	0	0
7	1	6	Feriado	0	5	15	0	0	0
8	1	7	Feriado	0	5	15	0	0	0
9	1	8	Feriado	0	5	15	0	0	0
10	1	9	Feriado	0	5	15	0	0	0
11	1	10	Feriado	0	5	15	0	0	0
12	1	11	Feriado	0	5	15	0	0	0
13	1	12	Feriado	0	5	15	0	0	0
14	1	13	Feriado	0	5	15	0	0	0
15	1	14	Feriado	0	5	15	0	0	0
16	1	15	Feriado	0	5	15	0	0	0
17	1	16	Feriado	0	5	15	0	0	0
18	1	17	Feriado	0	5	15	0	0	0
19	1	18	Feriado	0	5	15	0	0	0
20	1	19	Feriado	0	5	15	0	0	0
21	1	20	Feriado	0	5	15	0	0	0
22	1	21	Feriado	0	5	15	0	0	0
23	1	22	Feriado	0	5	15	0	0	0
24	1	23	Feriado	0	5	15	0	0	0
25	1	24	Feriado	0	5	15	0	0	0

▲	J	K	L	M	N	O	P
1	Carga_iluminacao_W	Carga_equipamentos_W	Carga_diversa_sensivel_W	Carga_diversa_latente_W	Carga_infiltracoes_sensivel_W	Carga_infiltracoes_total_W	Carga_infiltracoes_latente_W
2	13,125	56,25	0	0	-0,13729506	-0,198048843	-0,060753783
3	13,125	56,25	0	0	-0,14549178	-0,210903447	-0,065411667
4	13,125	56,25	0	0	-0,15163932	-0,21938732	-0,067748
5	13,125	56,25	0	0	-0,15676227	-0,227696621	-0,070934351
6	13,125	56,25	0	0	-0,15983604	-0,230591344	-0,070755304
7	13,125	56,25	0	0	-0,16290981	-0,236609681	-0,073699871
8	13,125	56,25	0	0	-0,16290981	-0,236609681	-0,073699871
9	13,125	56,25	0	0	-0,15881145	-0,231665945	-0,072854495
10	13,125	56,25	0	0	-0,14549178	-0,212588325	-0,067096545
11	13,125	56,25	0	0	-0,12704916	-0,188887407	-0,061838247
12	13,125	56,25	0	0	-0,10655736	-0,164933588	-0,058376228
13	13,125	56,25	0	0	-0,08811474	-0,142194398	-0,054079658
14	13,125	56,25	0	0	-0,07581966	-0,130509723	-0,054690063
15	13,125	56,25	0	0	-0,0717213	-0,125127213	-0,053405913
16	13,125	56,25	0	0	-0,0717213	-0,125127213	-0,053405913
17	13,125	56,25	0	0	-0,07479507	-0,128481043	-0,053685973
18	13,125	56,25	0	0	-0,07786884	-0,131937599	-0,054068759
19	13,125	56,25	0	0	-0,08299179	-0,137019564	-0,054027774
20	13,125	56,25	0	0	-0,08913933	-0,144246194	-0,055106864
21	13,125	56,25	0	0	-0,09631146	-0,151355129	-0,055043669
22	13,125	56,25	0	0	-0,10450818	-0,163098504	-0,058590324
23	13,125	56,25	0	0	-0,1127049	-0,172879856	-0,060174956
24	13,125	56,25	0	0	-0,12192621	-0,184795418	-0,062869208
25	13,125	56,25	0	0	-0,13012293	-0,194846699	-0,064723769

▲	Q	R	S	T	U	V
1	Carga_envolvente_externa_W_1	Carga_envolvente_externa_W_2	Carga_envolvente_externa_W_3	Carga_envolvente_externa_W_4	Carga_envolvente_externa_W_5	Carga_envolvente_externa_W_6
2	-86,8822	-37,8818	-137,685	-271,35	-273,36	-134
3	-94,1886	-40,1434	-145,905	-287,55	-289,68	-142
4	-98,1684	-41,8396	-152,07	-299,7	-301,92	-148
5	-101,4849	-43,2531	-157,2075	-309,825	-312,12	-153
6	-103,4748	-44,1012	-160,29	-315,9	-318,24	-156
7	-105,4647	-44,9493	-163,3725	-321,975	-324,36	-159
8	-105,4647	-44,9493	-163,3725	-321,975	-324,36	-159
9	-102,8115	-43,8185	-159,2625	-313,875	-316,2	-155
10	-94,1886	-40,1434	-145,905	-287,55	-289,68	-142
11	-82,2492	-35,0548	-127,41	-251,1	-252,96	-124
12	-68,9832	-29,4008	-106,86	-210,6	-212,16	-104
13	-57,0438	-24,3122	-88,365	-174,15	-175,44	-86
14	-49,0842	-20,9198	-76,035	-149,85	-150,96	-74
15	-46,431	-19,789	-71,925	-141,75	-142,8	-70
16	-46,431	-19,789	-71,925	-141,75	-142,8	-70
17	-48,4209	-20,6371	-75,0075	-147,825	-148,92	-73
18	-50,4108	-21,4852	-78,09	-153,9	-155,04	-76
19	-53,7273	-22,8987	-83,2275	-164,025	-165,24	-81
20	-57,7071	-24,5949	-89,3925	-176,175	-177,48	-87
21	-62,3502	-26,5738	-96,585	-190,35	-191,76	-94
22	-67,6566	-28,8354	-104,805	-206,55	-208,08	-102
23	-72,963	-31,097	-113,025	-222,75	-224,4	-110
24	-78,9327	-33,6413	-122,2725	-240,975	-242,76	-119
25	-84,2391	-35,9029	-130,4925	-257,175	-259,08	-127

	W	X	Y	Z	AA
1	Carga_envolvente_interna_W_1	Carga_envolvente_interna_W_2	Carga_envolvente_interna_W_3	Carga_sensivel_de_arrefecimento_W	Carga_latente_de_arrefecimento_W
2	56,688795	15,84	15,62085	-785,7716501	-0,060753783
3	56,688795	15,84	15,62085	-842,0878468	-0,065411667
4	56,688795	15,84	15,62085	-884,3249943	-0,067748
5	56,688795	15,84	15,62085	-919,5226173	-0,070934351
6	56,688795	15,84	15,62085	-940,641191	-0,070755304
7	56,688795	15,84	15,62085	-961,7597648	-0,073699871
8	56,688795	15,84	15,62085	-961,7597648	-0,073699871
9	56,688795	15,84	15,62085	-933,6016665	-0,072854495
10	56,688795	15,84	15,62085	-842,0878468	-0,067096545
11	56,688795	15,84	15,62085	-715,3764042	-0,061838247
12	56,688795	15,84	15,62085	-574,5859124	-0,058376228
13	56,688795	15,84	15,62085	-447,8744697	-0,054079658
14	56,688795	15,84	15,62085	-363,4001747	-0,054690063
15	56,688795	15,84	15,62085	-335,2420763	-0,053405913
16	56,688795	15,84	15,62085	-335,2420763	-0,053405913
17	56,688795	15,84	15,62085	-356,3606501	-0,053685973
18	56,688795	15,84	15,62085	-377,4792238	-0,054068759
19	56,688795	15,84	15,62085	-412,6768468	-0,054027774
20	56,688795	15,84	15,62085	-454,9139943	-0,055106864
21	56,688795	15,84	15,62085	-504,1906665	-0,055043669
22	56,688795	15,84	15,62085	-560,5068632	-0,058590324
23	56,688795	15,84	15,62085	-616,8230599	-0,060174956
24	56,688795	15,84	15,62085	-680,1787812	-0,062869208
25	56,688795	15,84	15,62085	-736,4949779	-0,064723769

	AB	AC	AD	AE
1	Carga_total_de_arrefecimento_W	Carga_sensivel_de_aquecimento_W	Carga_latente_de_aquecimento_W	Carga_total_de_aquecimento_W
2	-785,8324038	-943,2962951	-0,060753783	-943,3570488
3	-842,1532584	-999,6124918	-0,065411667	-999,6779034
4	-884,3927423	-1041,849639	-0,067748	-1041,917387
5	-919,5935516	-1077,047262	-0,070934351	-1077,118197
6	-940,7119463	-1098,165836	-0,070755304	-1098,236591
7	-961,8334647	-1119,28441	-0,073699871	-1119,35811
8	-961,8334647	-1119,28441	-0,073699871	-1119,35811
9	-933,6745209	-1091,126311	-0,072854495	-1091,199166
10	-842,1549433	-999,6124918	-0,067096545	-999,6795883
11	-715,4382424	-872,9010492	-0,061838247	-872,9628874
12	-574,6442886	-732,1105574	-0,058376228	-732,1689336
13	-447,9285494	-605,3991147	-0,054079658	-605,4531944
14	-363,4548647	-520,9248197	-0,054690063	-520,9795097
15	-335,2954822	-492,7667213	-0,053405913	-492,8201272
16	-335,2954822	-492,7667213	-0,053405913	-492,8201272
17	-356,414336	-513,8852951	-0,053685973	-513,938981
18	-377,5332926	-535,0038688	-0,054068759	-535,0579376
19	-412,7308746	-570,2014918	-0,054027774	-570,2555196
20	-454,9691012	-612,4386393	-0,055106864	-612,4937462
21	-504,2457101	-661,7153115	-0,055043669	-661,7703551
22	-560,5654535	-718,0315082	-0,058590324	-718,0900985
23	-616,8832349	-774,3477049	-0,060174956	-774,4078799
24	-680,2416504	-837,7034262	-0,062869208	-837,7662954
25	-736,5597017	-894,0196229	-0,064723769	-894,0843467